

THESIS / THÈSE

MASTER EN SCIENCES INFORMATIQUES

Conception de la base de connaissances d'un générateur de modèles d'aide à la décision

Petitjean, Therese

Award date:
1984

Awarding institution:
Universite de Namur

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Facultés Universitaires Notre-Dame de la Paix.

NAMUR

INSTITUT D'INFORMATIQUE.

CONCEPTION DE LA BASE
DE CONNAISSANCES D'UN GENERATEUR
DE MODELES D'AIDE A LA DECISION.

Mémoire présenté par

Thérèse PETITJEAN

en vue de l'obtention du
grade de licenciée et maître
en informatique.

année académique 1983 - 1984

A l'attention des lecteurs,

Je prie les lecteurs de bien vouloir
trouver ci-joints les feuilles d'errata, de même
qu'un complément de bibliographie.

En ce qui concerne les feuilles
d'errata, il faudra comprendre l..., même
ligne en partant du bas vers le haut, et l...,
même ligne en partant du haut vers le bas.

ERRATA.

- intervention des pages 41-42.
p 43 : ligne 9 colonne 6 : $E_9 \times F_9$.
- p. 52 : équation 1.
$$\text{Bénéfice}(\text{département} / \text{mois}) = \text{somme}(\text{produit}[\text{vente}])$$

département, bénéfice(département, produit / mois).
- p 53 : Modélisation de l'algorithme d'accès à
général.
4^{ème} ligne : $\text{FOR PRO} = \text{PRODUIT}[\text{VENTE}] \text{ DEP.}$
- p 54 (6) ligne 1 : "une demande de génération ...
un numéro de département".
- p 68 (24) ligne 4 : $\text{NBRE-UNITES- VENDUES- PAR- MOIS}$
- p 74 : L 8 : "... exclut sa participation ..."
- p 113 : L. 11 : "(respect de la contrainte citée p 99)".
- p 128 : L. 1 : "... de la relation "MODELE-DONNEE" = "D".
- p 135 : L. 7 : "... (si nécessaire) par un type de période."

p. 136 : L. 9 : exemple :

BENEFICE (DEPARTEMENT/MOIS) =
SOM (PRODUIT [VENTE] DEPARTEMENT ;
BENEFICE (PRODUIT/MOIS).

p. 146 : L. 10 : Attribut de ? $\leq T.E. \leq \text{mon } T.E.$ ou
 $\leq T.A. \leq \text{mon } T.A.$

p. 158 : L. 13 : COUT-ENBALLAGE (PRODUIT-FINI) =
(PRIX-ACHAT (: CONTENANT/AN) POUR...

ANNEXES

p. A1 : L. 5 : "... du monde réel qu'elle modélise"
L. 4 : "Une B.D. Histogramme peut donc être
p. considérée comme un ensemble de..."

p. A3 : L. 10 : "d'une association ..."

p. A8 : L. 4 : "Un intervalle de temps défini
sur une échelle de temps..."

p. A15 : L. 8, 9 : "Les occurrences produit i , date 1, date 2,
département k ,
produit i , date 3, date 4,
département k ,
sont différentes."

L. 4 : "Quels sont ..."

p. A 17 : L. 7 : "(date début. activité : D.D.A., date -
fin. activité : D.F.A.).

tableau : en tête de la dernière colonne :
D.F.A.

L. 12 : "Une D.F.A. inconnue ..."

L. 2 : "... déduire, à partir ..."

p. A 18 : L. 6 : "Ces attributs sont tels qu'une valeur
ne peut leur être attribuée ..."

p. A 20 : L. 1 : "Sur une échelle de temps de type
"mois" une période d'évaluation
de durée > jour doit être
représentée par deux dates.

- (1) JARNE T, SHALEV Jacob : A DATABASE ARCHITECTURE FOR SUPPORTING BUSINESS TRANSACTION, Journal of Management Information System, Spring 84.
- (2) H.R GUSTAFSSON, T. KARLSSON, J.A BUBENKO : A DECLARATIVE APPROACH TO CONCEPTUAL INFORMATION MODELING, Information systems design methodologies: a comparative review; Proc. of the IFIP TCS Work-Conf. 1982 North. HOLLAND Publ. Company.
- (3) CLIFFORD James (1982) : A LOGICAL FRAMEWORK FOR THE TEMPORAL SEMANTICS AND NATURAL LANGUAGE QUERYING OF HISTORICAL DATABASES; Ph. D. Dissertation, Dept of Computer Science, SUNY at STONY BROOK, STONY BROOK.
- (4) CLIFFORD James : A MODEL FOR HISTORICAL DATABASES; Working paper Series CRIS #47.
GBA # 32.76 (C.R.).
- (5) BRUCE B : A MODAL FOR TEMPORAL REFERENCE AND ITS APPLICATION IN A QUESTION ANSWERING PROGRAM; Artificial Intelligence (3) (1972).
- (6) A. BOLOUR and L.J. DEKEYSER : ABSTRACTION IN TEMPORAL INFORMATION; University of California, San Francisco, C.A. 94143 U.S.A. ; Infor. System vol. 8 n°1 1983.

- (7) David L. WALTZ : AN ENGLISH LANGUAGE QUESTION ANSWERING FOR A LARGE RELATIONAL DATABASE; University of Illinois at Urbana-Champaign. Communication of the ACM July 78, vol. 2 N7.
- (8) E. FALKENBERG : CONCEPTS FOR MODELLING INFORMATION; Modelling in D.B.D.S. ; éd. G. J. Nyssens - North. Holland. 76.
- (9) Bo SUNDGREN : CONCEPTUAL FOUNDATION OF THE INFOLOGICAL APPROACH TO DATA BASES; Proc. IFIP Conf. on Data Base management (1974) (p. 61-93).
- (10) Antoni OLIVE : DADES : A METHODOLOGY FOR SPECIFICATION AND DESIGN OF INFORMATION SYSTEMS; Informations design methodologies: a comparative review; T.W. Allen, H.G. Sel, A.A. Verryn. Stuart (Editors). North. Holland publ. Comp. IFIP, 1982.
- (11) W. KENT I.B.D. : DATA AND REALITY. Basic assumptions in data processing reconsidered; San Jose - California, North. Holland 78.

- (12) J. RUCHTI : DATA DESCRIPTION EMBEDDED IN CONTEXT ; MODELING IN DATA BASE MANAGEMENT SYSTEM (ed. Nyssen) North. Holland (76.)
- (13) JARKE Mathias (1982) : DEVELOPING DECISION SUPPORT SYSTEMS : A container / management Example; *International Journal of Policy Analysis and Information Systems* 6, 4 351-372.
- (14) R. STANPER, G. TAGY, P. NAJON, S. COOK : DEVELOPING THE LEGAL SEMANTIC GRAMMAR ; *Proc. Proceedings of the International Study Congress on Logical Information Systems* 6/10 April 1988 Firenze.
- (15) ARINN, GAD and HOWARD. L. MORGAN (1982) : EMBEDDING THE TIME DIMENSION IN INFORMATION SYSTEMS; T.R # 82-03.01 Dept. of Decision Sciences, The Wharton School, Univ. of Pennsylvania Philadelphia.
- (16) CLIFFORD, James and DAVID S. WAMER (1981) : FORMAL SEMANTICS FOR TIME IN DATABASES; T.R # 81/025 Dept of Computer Science, SUNY at STONY BROOK, Stony Brook.
- (17) L.A. BJORK J.R : GENERALIZED AUDIT TRAIL REQUIREMENTS; *IBM System Journal*, vol. 14 No. 3 p. 218.

(18) S. JONES and P. J. MASON : HANDLING THE TIME DIMENSION IN A DATA BASE; Proceedings International Conference on data Bases. Univ. of Aberdeen, July 80. Ed. S. D. Deen (univ. of Aberdeen and P. Hammersby (computer Journal)).

(19) Ricky OVERMYER and Michael STONEBRAKER : IMPLEMENTATION OF A TIME EXPERT IN A DATA BASE SYSTEM; Dept of electrical engineering and Computer science; Univ. of California Berkeley, Ca.

(20) J. A. RUBENKO : INFORMATION MODELING IN THE CONTEXT OF SYSTEM DEVELOPMENT; Proc. IFIP Congress 80 (Oct. 1980).

(21) Susan JONES, Peter MASON and Ronald STATPER : LEGOL 2.0 : A RELATIONAL SPECIFICATION LANGUAGE COMPLEX RULES; London School of Economics and Political Science, Houghton Street, London WC2A 2AE England. INFORMATION SYSTEM M. 4 Vol. 4 (1979) p293-305.

(22) S. COOK and R. STATPER : LEGOL AS A TOOL FOR THE STUDY OF BUREAUCRACY; The Information systems of Environment; ed. H. C. Lucas, Jr., J. J. Land, T. J. Lincoln and K. Supper. North-Holland.

- (23) JAMES F. ALLEN : MAINTAINING KNOWLEDGE ABOUT TEMPORAL INTERVALS ; Univ. of Rochester Comm. Act. November 73 V26 N1.
- (24) G. ARIAV and H.L. MORGAN : HDM : HANDLING THE TIME DIMENSION IN GENERALIZED DBMS. ; Working paper, Dpt of Decision Sciences, the Wharton School, Univ. of Pennsylvania (May 81).
- (25) V. DE ANTONELLIS and B. ZONTA : MODELLING EVENTS IN D.B. APPLICATION DESIGN PROCEEDINGS VERY LARGE DATA BASES (1971) ; Proc. 7th. V.L.D.B.
- (26) G.G. HENDRIX : MODELLING SIMULTANEOUS ACTIONS AND CONTINUOUS PROCESSES ; Artificial Intelligence (4) (1973).
- (27) T.L. ANDERSON : MODELING TIME AT THE CONCEPTUAL LEVEL ; Proceedings of the Second International Conference on databases (to appear), Jerusalem, Israel (June 82).
- (28) LANGEFORS B. : ON INFORMATION STRUCTURE AND DATA STRUCTURE ; (preprints IFIP-TCL-SPECIAL WORKING CONFERENCE: a technical in depth evaluation of the DDL - Belgium, Namur).

- (29) N. FINDLER and D CHEN: ON THE PROBLEMS OF TIME RETRIEVAL TEMPORAL RELATIONS, CAUSALITY, AND COEXISTENCE; Proc. 2nd Int. Joint Conf on Artificial Intelligence, Imperial College (1-3 sept. 1971).
- (30) JANIS A. RUBENKO Jr: ON THE ROLE OF UNDERSTANDING MODELS IN CONCEPTUAL SCHEMA DESIGN; Proceedings VLDB 5th int. conf. okt. 79.
- (31) J. BRADLEY: OPERATIONS DATA BASES; Proceedings of the fourth international conference on VLDB (Sept. 1978) (p164-176).
- (32) G. ARIAV, J. CLIFFORD, D. JARKE: PANEL ON TIME AND DATA BASES; A.C.M. Sigmod 13(4) mai 1973.
- (33) L. HIRSCHMAN and C. STONY: REPRESENTING EXPLICIT AND IMPLICIT TIME RELATIONS IN NARRATIVE; Proc. 7th. Int. Joint Conf. on Artificial Intelligence (p. 289. 295). Vancouver, British Columbia (August 1977).
- (34) G. WEDERHOLD, J.F. FRIES, S. WEGEL: STRUCTURED ORGANIZATION OF CLINICAL DATA BASES; Proc. AFIPS Nat. Conf. Conference (1975).

- (30)' A. SERNADAS : TEMPORAL ASPECTS OF LOGICAL PROCEDURE DEFINITION; London School of Economics Inform. Systems vol 5 n°3 1980.
- (31)' KLOPROGGE, MANFRED R. (1981) : TERM: AN APPROACH TO INCLUDE THE TIME DIMENSION IN THE ENTITY-RELATIONSHIP APPROACH; proc. 2nd Intern. Conf. on Entity Relationship Approach. Washington D.C.
- (32)' S.J.P. TODD : THE PETERLEE RELATIONAL TEST VEHICLE - A SYSTEM OVERVIEW; IBM Systems Journal Vol. 15(4) (1976) p295-308.
- (33)' A. BOLOUR, T.L. ANDERSON, L.J. DEKEYSER AND H.K.T. WONG : THE ROLE OF TIME IN INFORMATION PROCESSING : A SURVEY; ACM SIGMOD RECORD 12.3 (April 82).
- (34)' HAMMER Nicholl and Dennis McLEOD (1978) : THE SEMANTIC DATA MODEL : A MODELLING MECHANISM FOR DATA BASE APPLICATIONS; Proc. ACM SIGMOD 1978.
- (35) BUBENKO, JAVIS A. Jr (1977) : THE TEMPORAL DIMENSION IN INFORMATION MODELLING; Architecture and Models in data Base, management system, ed. G.T. Nyssens (North. Holland, Amsterdam).

- (36) B. BRENTMANN : THE TEMPORAL DIMENSION OF CONCEPTUAL SCHEMAS ; Proc. of the IFIP Working group 2.6. - Munich (March 1979)
- (37) LANGEFORS B. : THEORETICAL ANALYSIS OF INFORMATION SYSTEMS ; 4th ed. Studentlitteratur (Auerbach, , Sweden 1973).
- (38) LANGEFORS B. : THEORETICAL ASPECTS OF INFORMATION SYSTEMS FOR MANAGEMENT ; IFIP Congress 1974 (p. 937-945).
- (39) BO SUNDGREW : THEORY OF DATA BASES ; Petricelli Information System Series, New York, 1975.
- (40) EKARD FALKENBERG : THE HANDLING IN D.B.T.S. ; International C's Report 07/74.
- (41) F. JAKOB : UN EXEMPLE DE REPRESENTATION DE CONNAISSANCES SUR LE TEMPS ; Congrès Reconnaissance des formes et Intelligence artificielle , Univ. Paul. Sébastien. Toulouse.
- (42) SCHUELER Ben. Michael : UPDATE RECONSIDERED ; Architecture and models in Data Base Management Systems (éd. Nysen), North. Holland, 77.

- (43) ROLLAND C. AND C. RICHARD (1982) : TRANSACTION MODELLING ; Proc. ACM SIGMOD 1982,
- (44) A PRIOR : PAST, PRESENT AND FUTURE ;
Oxford Univ. Press (1967).
- (45) N. RESHER AND A. URQUHART : TEMPORAL LOGIC ;
Springer Verlag, N.Y. (1971).
- (46) R.P. TIL ARTHUR : TENSE LOGIC ;
REIDL, DORDRECHT (1976).
- (47) R. NATTISON : AN INTRODUCTION TO THE MODEL THEORY OF FIRST ORDER PREDICATE LOGIC AND A RELATED TEMPORAL LOGIC ; MEMO R.7-5570-1.PR, Rand Corporation, Santa Monica, CA (June 1969).
- (48) C.L. HANBLIN : INSTANTS AND INTERVALS ;
Proc. 1st Conf. Int. Soc. for the Study of Time ; ed. J.T. Fraser ; Springer Verlag (1972).
- (49) E. BENCI, F. BODART, H. BOGAERT, A. CABANES :
CONCEPTS FOR THE DESIGN OF CONCEPTUAL SCHEMA ; MODELLING IN DATA BASE MANAGEMENT SYSTEMS (ed. Nyssen) North-Holland 46.
187-193.

Je tiens à exprimer ma reconnaissance à Monsieur F. BODART, promoteur de ce mémoire. Je le remercie pour ses suggestions et ses conseils mais plus encore pour ses encouragements et sa compréhension.
Qu'il trouve ici l'expression de ma profonde gratitude.

Je remercie Madame V. Goemans pour l'assistance qu'elle m'a apportée dans la réalisation de ce travail.

Un grand merci à Marc ERVIER pour l'aide constante qu'il m'a prodiguée.
Qu'il trouve ici l'expression de mon amitié.

J'exprime également ma gratitude à Monsieur J.-C. COURBON pour l'accueil qu'il m'a réservé lors de mon séjour à Genève.

A Suzanne, pour les heures qu'elle a passées devant la machine à écrire

A Jean-Marie, pour le soin qu'il a apporté à la reproduction des schémas

A Marc et Marie-Anne, pour le dernier coup de main qu'ils m'ont donné: UN GRAND MERCI.

TABLE DES MATIERES

INTRODUCTION.....	1
PARTIE I : LES SYSTEMES INFORMATIQUES D'AIDE A LA DECISION.....	3
CHAPITRE I : INTRODUCTION AUX SYSTEMES INFORMATIQUES D'AIDE A LA DECISION.....	4
I.1. DEFINITION DES S.I.A.D.....	6
I.1.1. Les objectifs.....	6
I.1.2. Les personnes et les taches.....	7
I.2. CARACTERISTIQUES DES S.I.A.D.....	11
I.2.1. L'exploitation dynamique.....	11
I.2.2. L'"orientation-utilisateur".....	12
I.2.3. Le caractère évolutif des S.I.A.D.....	15
I.3. DEVELOPPEMENT DES S.I.A.D.....	17
I.3.1. Constatations empiriques.....	17
I.3.2. L'approche évolutive.....	18
I.3.3. Developpement "adaptatif".....	19
I.3.4. Schéma de l'application dynamique.....	21
CHAPITRE II : CARACTERISTIQUES GENERALES DES LOGI- CIELS D'AIDE A LA DECISION.....	25
II.1. TYPOLOGIE DES S.I.A.D.....	27
II.1.1. Critère de classification: complexité du support proposé.....	27
II.1.2. Critère de classification: généralité du système.....	28
II.2. ARCHITECTURE GENERALE DES S.I.A.D.....	31
II.2.1. Le sous-système des connaissances.....	31

II.2.2. Le sous-système des données.....	33
II.2.3. Le sous-système des modèles.....	34
II.2.4. Le sous-système du langage.....	35
II.2.5. Le sous-système de communication.....	35
CHAPITRE III : UN GENERATEUR DE TABLEAUX.....	37
III.1. OBJECTIF DU SYSTEME.....	37
III.2. DESCRIPTION D'UNE GENERATION DE TABLEAU.....	37
III.2.1. Les composants impliqués.....	39
III.2.2. Relation entre les différents composants.....	40
III.2.3. Exemple de generation de tableau.....	40
III.3. DESCRIPTION DE LA STRUCTURE GENERALE DU SYSTEME.....	46
III.3.1. L'application "exploitation et gestion" de la B.D. d'exploitation.....	46
III.3.2. L'application "gestion et exploitation de la B.D. de specification".....	50
III.3.3. L'application "génération-tableau".....	51
PARTIE II : ETUDE DU GENERATEUR DE MODELES.....	57
CHAPITRE I : DESCRIPTION DU SCHEMA CONCEPTUEL DE LA BASE DE CONNAISSANCES DU GENERATEUR.....	58
I.1. DESCRIPTION DU SCHEMA CONCEPTUEL DU SYSTEME D'INFORMATION.....	62
I.1.1. Description des éléments.....	62
I.2. DESCRIPTION DU SCHEMA CONCEPTUEL DU SYSTEME D'EQUATIONS.....	95
I.2.1. Definitions préliminaires.....	95
I.2.2. Description des éléments.....	96

I.2.3. Etudes d'exemples simples et conclusions.....	121
I.2.4. Modification du schéma conceptuel du système d'équations.....	125
I.3. INTEGRATION DES SOUS-SCHEMAS CONCEPTUELS DU SYSTE- ME D'INFORMATION ET DU SYSTEME D'EQUATIONS.....	130
I.3.1. Identité des éléments "type-donnée-disponible" et "attribut".....	130
I.3.2. Identité des éléments "arguments" et "type- entité" ou type-période".....	130
I.3.3. Description de la relation "correspondance" entre une "variable" et un "attribut".....	131
CHAPITRE II:: UN LANGAGE DE SPECIFICATION DE MODELES GENERAUX.....	133
II.1. CARACTERISTIQUE DU LANGAGE.....	133
II.2. Exemple 1.....	137
II.2.1. Explications.....	137
II.2.2. Système d'information.....	137
II.2.3. Système d'équations.....	139
II.2.4. Structures des tableaux.....	139
II.2.5. Execution du modèle.....	143
II.3. EXEMPLE 2.....	145
II.3.1. Explications.....	145
II.3.2. Système d'information.....	145
II.3.3. Système d'équations.....	148
II.3.4. Structures des tableaux.....	148
II.4. EXEMPLE 3.....	151

II.4.1. Explications.....	151
II.4.2. Système d'information.....	151
II.4.3. Systèmes d'équations.....	157
CONCLUSION.....	160
BIBLIOGRAPHIE.....	162
ANNEXE.....	165

I N T R O D U C T I O N

Les systèmes informatiques d'aide à la décision sont destinés à être exploités directement par les gestionnaires. Il est donc nécessaire de fournir à ces utilisateurs non informaticiens des procédés simples de communication avec le système.

Ces dernières années, nous avons assisté à une prolifération de logiciels d'aide à la décision de type "tableur".* Cependant, bien que relativement peu coûteux, ces logiciels ne rencontrent pas le succès auquel ils étaient promis.

Il est apparu que cet échec pouvait s'expliquer par le fait que les gestionnaires hésitent à utiliser des systèmes dont le langage (quoique relativement simple) leur semble rébarbatif.

C'est pourquoi nous avons envisagé la création d'un générateur de modèles d'aide à la décision qui permettrait aux gestionnaires d'utiliser un langage plus familier et qui leur éviterait une série de manipulations fastidieuses.

Ce système serait destiné à l'aide à la résolution de problèmes de décision, pour cette classe de problèmes pouvant être supportés par un logiciel de type "tableur".

Par le biais de ce mémoire, nous tenterons d'évaluer la complexité d'exploitation d'un tel système.

* Ex.: VISICALC, LOTUS, MICRO-MODELLER, IFPS...

La première partie de ce travail sera consacrée à l'étude générale des systèmes informatiques d'aide à la décision.

Le premier chapitre vise à donner une idée plus précise de ce qu'est un système informatique d'aide à la décision.

Dans le second chapitre, nous tenterons de donner un aperçu des technologies logicielles appropriées à ces systèmes.

Nous conclurons cette partie par la présentation du générateur de modèles d'aide à la décision que nous envisageons de réaliser.

La seconde partie sera consacrée à l'étude du générateur.

Le premier chapitre consistera en la description précise du contenu de la base de connaissances du générateur.

Le second chapitre sera consacré à une étude d'exemples d'utilisation d'un tel système. Nous y proposerons un langage de formulation d'équations générales.

L'annexe traitera du problème de la localisation temporelle des informations.

P A R T I E I

LES SYSTEMES INFORMATIQUES

D'AIDE A LA DECISION.

CHAPITRE I.

INTRODUCTION AUX SYSTEMES INFORMATIQUES D'AIDE A LA DECISION.

L'évolution des systèmes électroniques de traitement de données (Electronic Data Processing, E.D.P.) vers les systèmes informatiques de gestion (Management Information Systems, M.I.S.) a intégré l'informatique dans les organisations.

Bien que les systèmes informatiques jouissent d'un succès remarquable dans les organisations, ils n'ont été considérés longtemps que comme moyen d'accroître les performances dans l'exécution des tâches opérationnelles, en modifiant la manière dont l'information est gérée.

Ce n'est que vers les années 70 que leur utilisation potentielle dans les processus de prise de décision est apparue.

Ce nouveau champ d'applications a donné naissance au concept de "Système Informatique d'Aide à la Décision" (S.I.A.D)*, concept qui a fait couler beaucoup d'encre.

* Le terme généralement employé dans la littérature est "Decision Support System" (D.S.S.).

Le but de ce chapitre est de permettre au lecteur de se faire une idée générale des S.I.A.D. par la mise en évidence des différences fondamentales qui font qu'un S.I.A.D. se distingue des systèmes informatiques traditionnels que les organisations utilisent depuis des années.

La première partie du chapitre présentera une définition des S.I.A.D. Les implications de cette définition seront examinées plus profondément dans les parties 2 et 3 du chapitre. La partie 2 développera plus précisément les caractéristiques générales des S.I.A.D. Le processus de développement des S.I.A.D. sera présenté dans la troisième partie.

I.1. DEFINITION DES S.I.A.D.

Nous utiliserons la définition proposée par GINZBERG et STOHR(8, p. 12) :

" Un S.I.A.D. est un système informatique destiné à aider les gestionnaires à résoudre des problèmes de décision pour lesquels il n'est pas possible ou souhaitable d'appliquer un processus totalement automatisé ".

Cette définition met en évidence les différences fondamentales existant entre les systèmes informatiques d'aide à la décision et les systèmes informatiques de gestion traditionnels. Ceux-ci sont des systèmes destinés à accroître l'efficacité d'une organisation par l'accroissement de la rentabilité du personnel d'exécution, grâce à l'amélioration des procédures de traitement et de gestion de l'information.

L'analyse de ces définitions permet de faire une distinction entre ces types de systèmes au niveau:

- des objectifs en fonction desquels ils sont conçus.
- des personnes auxquelles ils s'adressent.
- des types de tâches qu'ils sont amenés à effectuer.

I.1.1. LES OBJECTIFS.

↳ Cette différence est fondamentale: elle limite à elle seule le champ des S.I.A.D.

Ce n'est pas paraphraser que de souligner que l'unique objectif des S.I.A.D. est "l'aide à la prise de décision". Ceci implique l'INTEGRATION de l'outil informatique dans une DEMARCHE DE GESTION.

Un S.I.A.D. est donc une aide, pas un substitut: il acquiert une fonction comparable à celle d'un consultant.

Il sera donc introduit dans une organisation dans le but d'accroître l'efficacité de l'organisation: meilleurs choix stratégiques, meilleure allocation des ressources, meilleur contrôle de l'allocation des ressources,...

Par contre, l'objectif des S.I. traditionnels est l'accroissement de l'efficience de l'organisation: augmentation du volume de travail effectué, réduction des coûts d'exploitation.

Bien que pouvant parfois produire des informations utiles à un processus de décision (ex.: comptabilité), ces systèmes d'information ne peuvent être considérés comme des S.I.A.D. Ils ne sont introduits que dans le but d'accélérer ou faciliter des tâches opérationnelles; la production de rapports ou le stockage d'informations ne doivent être considérés ici que comme des sous-produits et non pas comme des "fonctions S.I.A.D.".

Par contre, la possibilité offerte à un gestionnaire de pouvoir, à partir de son poste de travail, accéder et exploiter des données enregistrées, formater des rapports, à l'aide d'un langage simple, constituerait une "fonction S.I.A.D.".

I.1.2. LES PERSONNES ET LES TACHES.

Ces deux aspects différenciant un S.I.A.D. d'un S.I.G. (système informatique de gestion) ne sont en fait que la conséquence des objectifs attribués à ces deux types de systèmes et discutés précédemment.

Les types de tâches prises en charge par les S.I.G. peuvent être qualifiées de tâches " STRUCTUREES ", c'est-à-dire de pure exécution, basées sur un ensemble de règles et de procédures.

Bien que ce type de tâche nécessite une certaine compétence et des connaissances particulières, elles ne font pas appel au

jugement. Les personnes assignées à ce genre de travail ont un statut d'exécutant.

Par contre, les tâches relevant du domaine des S.I.A.D. sont qualifiées de " SEMI-STRUCTUREES ". Ce concept a été introduit par GORRY et MORTON (9) pour caractériser les tâches qui ne sont ni "structurées" (routinières et automatisables car ne nécessitant pas le jugement), ni "non-structurées" (ne faisant intervenir que le jugement humain, donc défiant toute automatisation).

Un schéma conceptuel du processus de prise de décision a été proposé par H. SIMON. Il est décrit en termes des étapes le composant:

- 1. INFORMATION (identification de l'environnement concerné par le problème de décision):
recherche, traitement et analyse de données, de manière à obtenir des indications permettant d'identifier le problème.
- 2. ETUDE (développement et analyse des différentes alternatives possibles):
cette étape nécessite des capacités de compréhension des problèmes, de génération de solutions, et d'évaluation de solutions.
- 3. CHOIX: une solution particulière est privilégiée et implémentée.

Les problèmes de décision pour lesquels "il n'est pas possible ou souhaitable d'appliquer un processus totalement informatisé" (cfr supra 1.1.) sont les problèmes semi-structurés définis par KEEN et MORTON (6, p. 95) comme des problèmes dans lesquels une ou deux (mais pas trois) phases du processus de décision présentées dans le modèle de SIMON ne sont pas structurées.

Cette définition n'est peut-être pas suffisamment formelle puisque parfois les organisations décident de traiter des problèmes semi-structurés comme s'ils étaient structurés; le degré de structuration attribué à un problème n'est donc qu'une question de perception.

Elle est cependant acceptable car intuitivement convaincante: elle met en évidence le rôle qu'ont les S.I.A.D. dans l'AIDE au jugement, non pas en supportant tout le processus de décision, mais en fournissant au décideur un ensemble de fonctions qu'il pourra utiliser et mettre en oeuvre à chaque stade du processus de décision, tout en respectant son style cognitif.

La figure 1. présente les différentes phases du processus de décision définies dans le modèle de SIMON.

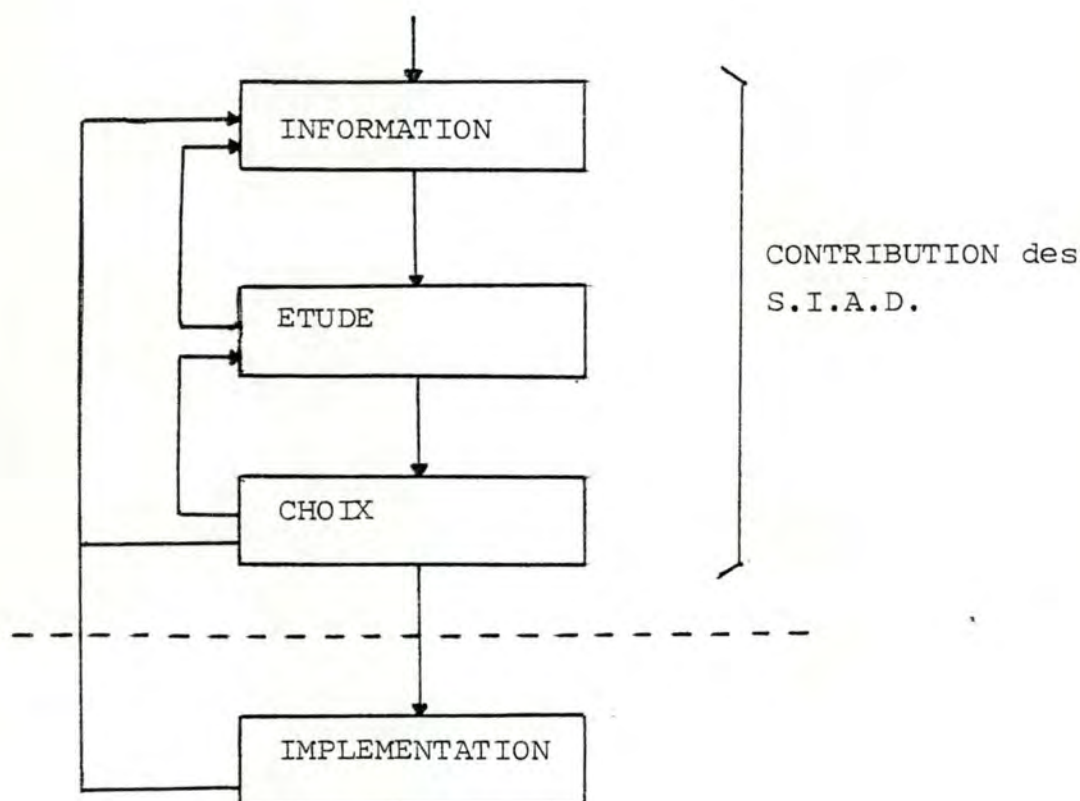


figure 1. : modèle de SIMON.

CONCLUSIONS

De la définition des S.I.A.D., il ressort que:

- Les S.I.A.D. ne sont pas des systèmes résultant de l'amélioration et de l'évolution "naturelle" des systèmes informatiques traditionnels: ils ne les remplaceront probablement pas.

- Leur apparition ne résulte pas de la volonté d'introduire à tout prix l'informatique dans des postes de "top-management", mais de l'opportunité d'une situation: l'existence d'une certaine classe de problèmes et d'un certain type de technologie.

- Le défi lancé par les S.I.A.D. est de créer une symbiose entre ces problèmes et ces technologies, entre l'homme et l'outil informatique, par la mise en commun de l'intuition, du jugement, de la personnalité du premier et des possibilités de traitement et de mémoire du second.

Ce sont donc des systèmes intégrables dans une démarche de décision, exploitables par les décideurs et destinés à améliorer leur efficacité au sein d'une organisation.

I.2. CARACTERISTIQUES DES S.I.A.D.

Bien que théorique, cette introduction aux S.I.A.D. peut permettre d'en déduire intuitivement les caractéristiques générales.

En premier lieu, relevons le fait que les S.I.A.D. sont tout d'abord des systèmes d'information. Ils sont donc composés:

- d'informations: représentation des faits importants pour l'utilisateur.
- de processus d'acquisition, de stockage, et de communication de ces informations.

Dégageons à présent les caractéristiques plus spécifiques à ce type de système d'information:

- un S.I.A.D. doit être exploitable dynamiquement
- un S.I.A.D. doit être "orienté - utilisateur" ("user-friendly").
- un S.I.A.D. est amené à évoluer.

I.2.1. L'EXPLOITATION DYNAMIQUE.

C'est volontairement que nous n'avons pas utilisé l'expression "utilisation interactive" qui pourrait laisser supposer que l'utilisateur reste passif et qu'il n'a plus la possibilité d'intervenir dans l'évolution du processus de décision, une fois qu'il a été pris en charge par l'outil informatique.

L'expression "exploitation dynamique" met en évidence le rôle actif joué par l'utilisateur qui garde le contrôle du processus

de décision. Celui-ci se déroule grâce à l'interaction (continue ou non) de l'homme et de l'outil.

Cette interaction est particulière, en ce sens que l'outil SUBIT l'action du décideur, tandis que le décideur REAGIT à l'action de l'outil.

Cette distinction est importante car "l'interactivité" n'est pas une caractéristique particulière aux S.I.A.D. (les utilisateurs des systèmes informatiques traditionnels utilisent également le système de manière interactive). C'est plutôt le caractère dynamique de son exploitation qui le distingue des autres systèmes.

I.2.2. L'"ORIENTATION - UTILISATEUR".

Etant intégré dans sa démarche et faisant partie de son poste de travail, un S.I.A.D. est destiné à être exploité par le gestionnaire.

Une attention particulière doit donc être accordée aux processus de communication décideur-système (et système-décideur).

La figure 2 (10, P. 45) est une représentation schématique d'un "cycle-communication".

- "Thought" fait référence à l'interprétation, par l'utilisateur, des données affichées.
- "Action plans" fait référence à la modification que l'utilisateur désire effectuer sur cet affichage (ceci est évidemment dépendant des possibilités offertes par le système).
- "Translate for system" fait référence aux étapes par lesquelles l'utilisateur doit passer pour transformer l'action désirée en actions acceptées par le système.

Le processeur du système complète la boucle en interprétant les actions de l'utilisateur, en recherchant et en traitant des

données si nécessaire, et en présentant le résultat dans une forme compréhensible pour l'utilisateur.

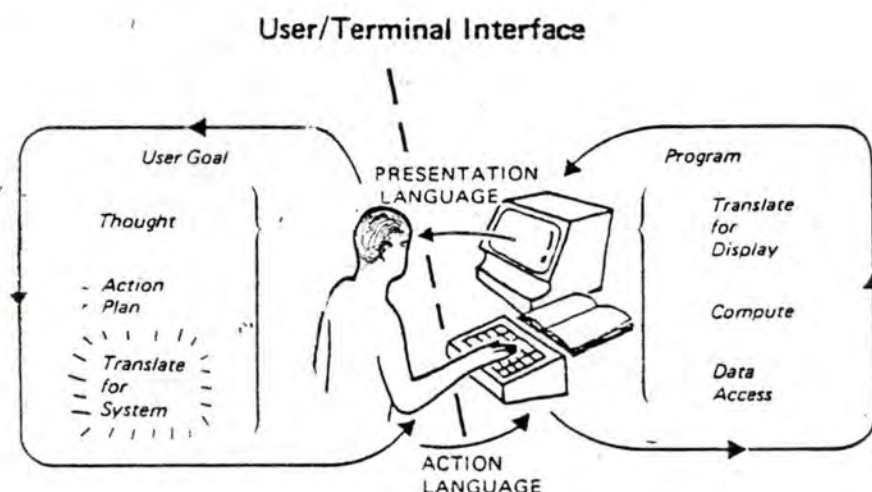


figure 2. : cycle-communication.

Les trois étapes représentées dans la "boucle utilisateur" sont en fait interdépendantes et étroitement mêlées dans son processus de raisonnement.

Schématiquement séparées, elles nous permettent de mieux discerner les faits à prendre spécialement en considération dans le contexte d'un système directement exploitable par le décideur.

a) l'utilisateur n'est pas informaticien: il doit communiquer avec le système, sans avoir à connaître la programmation.

Le dialogue sera rendu possible grâce à l'utilisation de langages de haut niveau, tels que:

- menus, systèmes question-réponse, écrans formatés à compléter, ... (dialogues conduits par le système)
- langage de commande, langage de "query", langage "naturel", ... (dialogues conduits par l'utilisateur).

Citons également les langages de manipulation d'objets (11, 12, 13, 14).

Pour fournir à l'utilisateur un procédé de communication "simple et naturel" ("simple et naturel" signifiant: facile à apprendre, facile à utiliser, conduisant rarement à l'erreur, nombre d'étapes réduit - "translate for system" minimal -, ...) plusieurs techniques de dialogues pourront coexister, la présentation servira de support à l'action (ex.: langage de commande présenté sous forme de menu).

b) l'utilisateur a des techniques de travail personnelles:

Le S.I.A.D. doit être personnalisé: la présentation doit être conforme ou proche des techniques de travail habituellement utilisées par le gestionnaire (utilisation de graphiques, tableaux, équations, ...).

c) l'utilisateur a besoin de comprendre la logique du système:

Des résultats fournis par le système ne seront acceptés par l'utilisateur que s'il comprend comment ils ont été calculés: selon quelle méthode et à partir de quelles données.

Les décideurs se méfieront de systèmes "boîte noire" trop complexes. Il sera donc parfois nécessaire que le système puisse "expliquer" son raisonnement.

I.2.3. LE CARACTERE EVOLUTIF DES S.I.A.D.

KEEN considère cette caractéristique suffisamment significative que pour l'admettre comme seul critère d'identification des S.I.A.D.:

"The label "Support System" is meaningful only in situations where de "final" system emerge through an adaptive process of design and usage" (15, p. 28).

A cette définition, nous rétorquerons que "si le système "final" avait été fourni tel quel à l'utilisateur, il n'en serait pas moins un S.I.A.D.":

Pour dégager ce qui caractérise effectivement les S.I.A.D., confrontons le jugement de GINZBERG et STOHR à la définition de KEEN.

"Usage patterns for all systems, not just DSS, evolve. Additional data items are placed in an employee master file and the payroll system becomes a personnel system... Evolutionary usage is not unique to DSS... There seems to be some confusion here between DSS and novelty...

...Perhaps some DSS will evolve more rapidly than other systems because they start with less structuring and thus have more "room" for change.

... In summary, evolutionary usage does not appear to be a particularly useful way to characterize DSS. It fails to distinguish DSS from other computer-based systems, and ascribes to all DSS something wich is characteristic only of some." (8, p.21).

Cette confrontation d'opinions met en évidence que ce qui différencie ces deux types de systèmes n'est pas leur propension à l'évolution, mais leur processus d'évolution, les stimulants de l'évolution:

- L'évolution des systèmes présentée par GINZBERG et STOHR est une évolution normale et prévisible: la prise en charge de tâches opérationnelles (structurées), de plus en plus nombreuses par l'outil informatique (extension "normale" d'un système).
- Par contre, et c'est ce que suggère KEEN dans sa définition, l'évolution d'un S.I.A.D. est la conséquence de l'intégration de l'outil dans une démarche non structurée et conduite par l'utilisateur. Pour mener à bien le processus de décision, l'utilisateur PERSONNALISE l'outil. La manière dont le système va évoluer est a priori imprévisible (extension "personnalisée" d'un système).

La question qui se pose donc est de savoir s'il est possible ou même souhaitable de construire un S.I.A.D. "produit fini". Cette question dépasse le cadre des caractéristiques des S.I.A.D., elle sera débattue dans la partie suivante.

CONCLUSIONS.

Devant être intégré dans une démarche conduite par des non-informaticiens, un S.I.A.D. doit être exploitable dynamiquement et de manière aisée. Il doit également être suffisamment personnalisé que pour être "attractif".

Le caractère non structuré d'un processus de décision et la personnalité de l'utilisateur sont des facteurs propices à son évolution qui est, a priori, imprévisible.

I.3. DEVELOPPEMENT DES S.I.A.D.

Le type de tâches que les S.I.A.D. sont amenés à supporter, ainsi que la particularité de leur caractère évolutif, montrent que le concept de "S.I.A.D." est un concept nécessaire.

Cette constatation conduit à une question:

"Un S.I.A.D. est-il à ce point particulier que pour nécessiter un processus de développement différent de celui utilisé pour les S.I. traditionnels ?"

Une réponse valable à cette question ne peut être fondée sur des arguments théoriques. Des S.I.A.D. ont déjà été implémentés et utilisés. Nous pouvons donc nous baser sur les constatations empiriques, l'expérience et les travaux des spécialistes S.I.A.D..

I.3.1. CONSTATATIONS EMPIRIQUES.

L'étude et l'analyse de S.I.A.D. particuliers, effectuées notamment par KEEN (A review of DSS - Case Studies), ont permis, entre autres, de mettre en évidence les points suivants:

- L'usage des S.I.A.D. est généralement différent de l'usage prévu. De plus, les utilisations les plus valorisantes et innovatives n'auraient pu être prédites quand le système était à l'étude.
- L'utilisation est personnalisée: un décideur exploite le S.I.A.D. selon son propre style.
- Un S.I.A.D. évolue; on constate qu'un des facteurs favorables à une évolution réussie est une architecture flexible qui permet des modifications et des extensions rapides.
- Le succès d'un S.I.A.D. dépend de son évolution plutôt que de son utilisation initiale.

I.3.2. L'APPROCHE EVOLUTIVE.

Les études relatives au cycle de vie d'un projet informatique (traditionnel) ont mis en évidence la nécessité d'une méthodologie de développement: analyser, spécifier et concevoir, puis seulement programmer, tester et installer. BROOKS(16) a établi que la programmation ne représentait que 10 à 20% de l'effort de développement.

Par contre, les différents rapports établis sur base des recherches effectuées sur les systèmes relevant du domaine des S.I.A.D., recommandent vivement l'utilisation d'une autre stratégie de développement: L'APPROCHE EVOLUTIVE.*

L'approche évolutive est une méthodologie basée sur la conception progressive d'un système à travers de multiples cycles "analyse-conception-mise en oeuvre-utilisation-évaluation", aussi courts que possible, à l'occasion desquels les versions successives du système en construction sont utilisées par l'utilisateur final. (19, p. 6)

Cette méthodologie diffère donc principalement des techniques traditionnelles par le fait qu'au départ, on travaille sur base de spécifications fonctionnelles partielles:

- l'information est pauvrement structurée.
- tous les faits ne sont pas représentés par l'information.
- les messages "entrée" et "sortie" sont mal définis.

* Ce concept a été introduit par NESS sous l'appellation "middle out design". Ses rapports et études de cas (2) et (6) donnent un aperçu du développement de ce concept.

Le terme "approche évolutive" a été introduit par JOHNSON, COURBON, OUDET (17). L'application de cette méthodologie dans le contexte des S.I.A.D. a été plus particulièrement étudiée dans (18, 19, 20).

- l'enchaînement des traitements est mal connu.
- les règles de traitement sont partiellement connues.

Ce n'est que grâce à son évolution à travers les différents cycles que le système pourra être spécifié complètement.

L'approche évolutive suggère donc une modification de la méthodologie traditionnelle qui consiste à déplacer la concentration de l'effort de développement :

- ne pas retarder la programmation, mais réaliser cette étape le plus rapidement possible.
- ne pas chercher à définir clairement un système "final" implémentable, mais implémenter un système "initial" qui peut être modifié et qui peut évoluer.

"The systems development life cycle is a strategy for finishing: adaptive design (a term that captures all the middle-out, incremental, and evolutionary techniques scattered throughout the case studies) is a method for beginning". (15, p. 27)

Les constatations citées précédemment justifient cette approche.

I.3.3. DEVELOPPEMENT "ADAPTATIF".

Remarquons que l'approche évolutive, la nature semi-structurée des tâches qu'un S.I.A.D. doit supporter, et le caractère imprévisible de son utilisation, expliquent la définition de KEEN (cfr supra 2.3.).

Un développement "adaptatif" peut être nécessaire pour l'une ou l'autre des raisons suivantes :

- 1) le concepteur ou l'utilisateur ne peut pas fournir de spécifications fonctionnelles ou ne veut pas le faire; soit qu'il n'arrive pas à

faire correspondre des procédures aux besoins (le degré de structure d'une tâche dépend de la perception), soit qu'il sent qu'une telle correspondance est impossible (le manque de structure est intrinsèque à la tâche).

- 2) les utilisateurs ne savent pas très bien ce qu'ils veulent et les concepteurs ne peuvent comprendre ce dont ils ont besoin ou ce qu'ils peuvent accepter; un système initial doit être construit pour donner à l'utilisateur un outil concret qui le poussera à réagir (c'est l'idée sous-jacente au concept d'"approche évolutive").
- 3) le S.I.A.D. influence la perception que l'utilisateur a d'une situation de décision. Le système stimule l'étude; ce qui, en retour, stimule l'utilisation de nouveaux procédés et la nécessité d'ajouter de nouvelles fonctions au système.
Le caractère non prévisible de l'usage d'un S.I.A.D. est le reflet de cette situation qui ne peut être exploitée que si le S.I.A.D. évolue en fonction des besoins.
- 4) les utilisateurs potentiels du système sont autonomes et les différentes méthodes qu'ils emploient pour mener à bien la tâche sont un obstacle à la standardisation.

Remarquons que 3) suggère que le S.I.A.D. peut "modeler" l'utilisateur, tandis que 1), 2), 3) et 4) suggèrent que l'utilisateur peut également "modeler" le système.

Il ressort de ces constatations que concevoir un S.I.A.D. comme un système traditionnel peut être une erreur.

En effet, une évolution qui ne se ferait que grâce à l'inter-

action de l'utilisateur et du concepteur est une évolution "non créative": le décideur utilise et s'adapte.

En revanche, la prise en considération de l'interaction S.I.A.D.-utilisateur permet de "construire réellement un S.I.A.D., pas un modèle" (15, p. 25) : le décideur exploite et adapte.

I.3.4. SCHEMA DE L'ADAPTATION DYNAMIQUE.

Une meilleure compréhension de la dynamique des relations d'adaptation entre les différents acteurs (outil, concepteur, utilisateur) est donc nécessaire au bon développement d'un S.I.A.D.

KEEN modélise ces interrelations par le schéma représenté par la figure 3 (15, p. 29). Ce schéma constitue en fait la synthèse des différents points présentés dans cette partie.

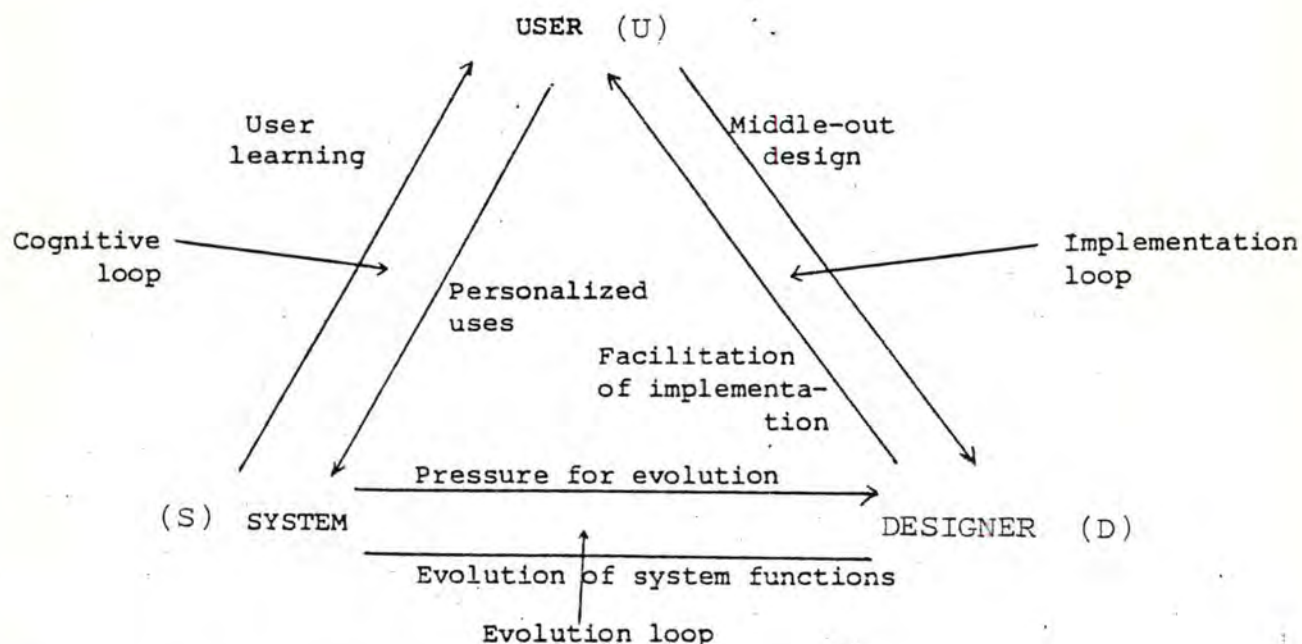


Figure 3.: adaptation dynamique.

Chaque flèche représente une direction d'influence.

I.3.4.1. LA BOUCLE COGNITIVE.

De nouveaux outils doivent promouvoir de nouvelles utilisations et vice-versa.

Le S.I.A.D. doit être conçu pour stimuler le raisonnement de l'utilisateur (relation " $S \rightarrow U$ ").

Comme l'utilisateur est susceptible de développer une nouvelle approche à la résolution du problème, il ne doit pas être contraint par la précédente.

Si le système fournit une séquence rigide de procédures, cette stimulation de l'évolution est bloquée. La relation " $U \rightarrow S$ " est basée sur une architecture qui permet une utilisation personnalisée, afin que les modifications de raisonnement de l'utilisateur puissent être (directement) exploitées.

La boucle cognitive est encore mal connue: ce n'est que depuis peu de temps qu'elle est objet de recherche.

Les recherches concernant la relation " $S \rightarrow U$ " consistent essentiellement en l'analyse des caractéristiques des problèmes et systèmes, ayant conduit à une utilisation personnalisée non prévisible.

Les recherches relatives à la relation " $U \rightarrow S$ " consistent en l'étude psychologique des différences individuelles.

Ces travaux ne peuvent être menés de manière indépendante.

I.3.4.2. LA BOUCLE D'IMPLEMENTATION.

La relation " $U \rightarrow D$ " montre que l'approche évolutive est un moyen, pour le concepteur, de mieux comprendre les désirs et les besoins de l'utilisateur.

Le processus d'implémentation nécessite de la part du concepteur, une stratégie qui dépasse l'analyse traditionnelle.

La relation " $D \rightarrow U$ " met en évidence la particularité de cette stratégie: avant d'être un produit, un S.I.A.D. doit être un service.

Ayant fait l'objet de nombreux travaux (réalisés notamment par COURBON, cfr supra I.3.2.), la boucle d'implémentation est mieux connue que la précédente.

I.3.4.3. LA BOUCLE D'EVOLUTION.

La boucle d'évolution se rapporte au processus par lequel l'influence du S.I.A.D. sur le raisonnement du décideur, l'utilisation personnalisée, l'approche évolutive et la stratégie du concepteur se combinent pour rendre nécessaire l'évolution du système: de nouvelles fonctions doivent être ajoutées (relation " $D \rightarrow S$ ").

Connaître quand et comment faire évoluer le système (indépendamment d'une demande de l'utilisateur) est une fonction du concepteur.

La relation " $S \rightarrow D$ " représente les connaissances que le concepteur peut acquérir grâce au système. Ceci peut être fait, par exemple, en gardant trace de l'utilisation du système par le décideur (commandes et séquences les plus utilisées, étude des différentes méthodes d'exploitation d'un même système par des utilisateurs différents, ...).

CONCLUSIONS.

C'est principalement à partir de constatations empiriques qu'a été mise en évidence la nécessité d'utiliser, pour les S.I.A.D., un processus de développement particulier: concevoir "un" système initial pour produire "le" système final.

Ce processus doit être conduit de telle manière que le système "final" soit le résultat de relations mutuelles d'influence et d'adaptation entre 3 acteurs: le concepteur, le système et l'utilisateur.

Sur un plan pratique, cette stratégie se traduit par: "construire, modifier et étendre constamment".

Ce n'est possible que si:

- le concepteur possède des qualités telles que la perspicacité et l'intuition.
- l'architecture du système est modulaire, flexible et extensible.
- le programmeur a la possibilité d'implémenter rapidement de nouvelles fonctions (ou de modifier facilement d'anciennes).

Ceci souligne l'importance à accorder au développement des techniques de structuration de logiciel et de programmation structurée, ainsi qu'à la mise au point de nouveaux langages. Ceux-ci devraient permettre, non seulement d'écrire rapidement de nouveaux composants, mais aussi de combiner facilement des logiciels existants.

CHAPITRE II.

CARACTERISTIQUES GENERALES DES LOGICIELS D'AIDE A LA DECISION.

Les recherches effectuées dans le domaine de l'aide à la décision se sont tout d'abord dirigées vers les problèmes liés aux comportements et aux besoins des utilisateurs (interface homme-machine, impact des S.I.A.D. sur les personnes, méthodologie de conception intégrant la participation de l'utilisateur,...).

Afin de se donner une image relativement complète des S.I.A.D., il est également nécessaire de considérer ces systèmes d'un point de vue technologique:

- les recherches relatives aux comportements et besoins des utilisateurs déterminent ce qui doit être fait.
- les progrès technologiques déterminent ce qui peut être fait.

Les aspects "besoins" et "technologie" sont donc fortement liés. Les recherches relatives au premier devraient, idéalement, promouvoir le progrès du second.

Ce chapitre a pour but de donner un aperçu des technologies logicielles appropriées aux S.I.A.D.

Nous proposerons tout d'abord une typologie des S.I.A.D., présentée selon deux critères:

- la complexité du support fourni au gestionnaire.
- la généralité du système proposé.

Nous décrirons ensuite une architecture générale présentant grossièrement les différents constituants logiciels d'un S.I.A.D.

II.1. TYPOLOGIE DES S.I.A.D.

II.1.1. CRITERE DE CLASSIFICATION: COMPLEXITE DU SUPPORT PROPOSE.

Ce critère nous conduit à distinguer trois types de S.I.A.D.:

a) Les S.I.A.D. procurant un ensemble de fonctions simples telles que:

- la gestion de fichiers
- le formattage de rapports
- l'utilisation de fonctions statistiques
- la construction de graphiques
- la manipulation de tableaux de données
- l'interrogation d'une base de données

Ces systèmes fournissent donc un support à des aspects ponctuels du processus de raisonnement.

Puisqu'ils procurent un ensemble de fonctions couramment utilisées dans un environnement de gestion, les S.I.A.D. de ce type peuvent être considérés comme les systèmes bureautiques des gestionnaires.

b) Les S.I.A.D. fournissant à l'utilisateur la possibilité d'exécuter des modèles complexes tels que:

des modèles économétriques, des modèles prévisionnels, des modèles d'optimisation, ... à partir de données stockées dans une B.D.

Ces systèmes sont basés sur une formalisation poussée du processus de raisonnement du gestionnaire (problèmes relativement structurés).

c) Les systèmes experts issus de l'application des techniques développées en intelligence artificielle.

Grâce à l'accumulation structurée de connaissances (spécifiques à un certain domaine), ces systèmes atteignent des performances comparables à celles d'un expert humain.

Les fonctions S.I.A.D. offertes par les systèmes experts sont :

- l'établissement de diagnostics
- la suggestion de jeux d'alternatives
- l'évaluation des alternatives

ainsi que la justification de leur processus de raisonnement.

Les problèmes traités par de tels systèmes sont relativement peu structurés.

II.1.2. CRITERE DE CLASSIFICATION: GENERALITE DU SYSTEME.

Selon la généralité de l'outil proposé, nous pouvons également distinguer deux grandes classes de S.I.A.D. (9):

a) Les S.I.A.D. spécifiques construits pour aider le gestionnaire dans la résolution de problèmes (ou classes de problèmes) particuliers.

Ces systèmes supportent donc des "applications S.I.A.D.".

b) Les générateurs S.I.A.D.

Un générateur S.I.A.D. est un logiciel fournissant la possibilité de construire facilement un S.I.A.D. spécifique.

La complexité d'un générateur S.I.A.D. réside essentiellement dans la création d'un interface commun à des fonctions individuellement utilisables, afin de pouvoir travailler sur des ensembles communs de données.

Exemples:

- .GADS (Geodata Analysis and Display System) (21)
- .EIS (Executive Information System) décrit dans (22,p.9) comme étant un ensemble d'outils intégrés, fournissant la possibilité d'utiliser des routines d'analyse financière et statistique, de préparer des rapports, d'interroger une B.D., d'afficher des graphiques. Il peut donc être utilisé comme générateur, dans le cadre de problèmes de décision financière.

Nous pouvons donc décrire un générateur S.I.A.D. comme un système de développement d'applications pour cette classe d'applications que nous appelons S.I.A.D. spécifiques.

D'un point de vue technologique, il est également important de considérer ce que nous appellerons les "outils S.I.A.D.".

Ce sont les éléments permettant, soit de développer directement une application S.I.A.D., soit de développer un générateur S.I.A.D.

La relation entre les différents niveaux technologiques selon lesquels un S.I.A.D. peut être considéré est représenté dans la figure 4. (22, p.10).

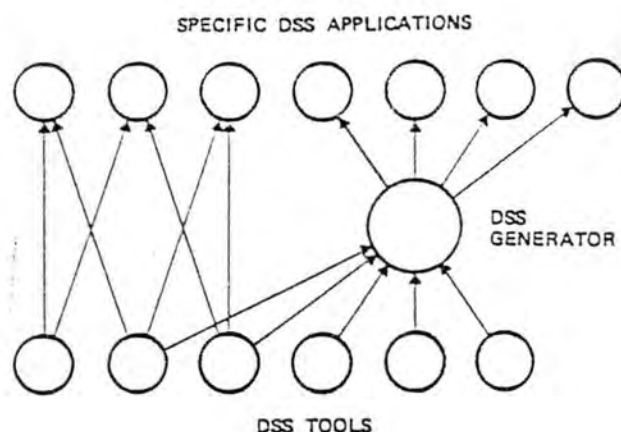


Figure 4.

1) Des outils de base peuvent être utilisés directement pour développer une application S.I.A.D. Cette approche est identique à celle utilisée pour le développement d'applications traditionnelles (utilisation d'outils comme: langages de programmation, S.G.B.D., logiciels graphiques, ...)

Ces outils n'étant généralement pas "orientés non informaticiens", cette approche restreint la participation de l'utilisateur dans la construction, la modification et l'extension de son S.I.A.D.

2) Les générateurs S.I.A.D. ont comme objectif de minimiser le temps nécessaire à la création d'une "application S.I.A.D." spécifique et d'encourager la participation réelle de l'utilisateur à cette création.

Les utilisateurs pourront donc se servir d'un générateur pour construire un ensemble d'applications appropriées

à leurs problèmes de décision.

Cependant, étant donné l'état actuel de la technologie et l'attitude de la plupart des décideurs, les utilisateurs de générateurs seront généralement des "intermédiaires" ayant une bonne connaissance des problèmes du gestionnaire ainsi qu'une certaine facilité de compréhension des possibilités d'un système informatique.

II.2. ARCHITECTURE GENERALE DES S.I.A.D.

Les composants principaux des logiciels S.I.A.D. sont représentés par la figure 5.

Cette architecture est générale: aucun S.I.A.D. connu ne contient tous les composants décrits.

Il est bon de remarquer que cette architecture représente chaque constituant comme étant une unité indépendante possédant donc des fonctions et des caractéristiques propres.

II.2.1. LE SOUS-SYSTEME DES CONNAISSANCES.

La base de donnée des connaissances contient le schéma conceptuel du système: la spécification des données manipulées et des modèles utilisés.

La base de connaissances a une fonction essentiellement documentaire:

- pour l'utilisateur, en lui fournissant:
 - . la sémantique des données.
 - . la définition et l'explication des modèles.

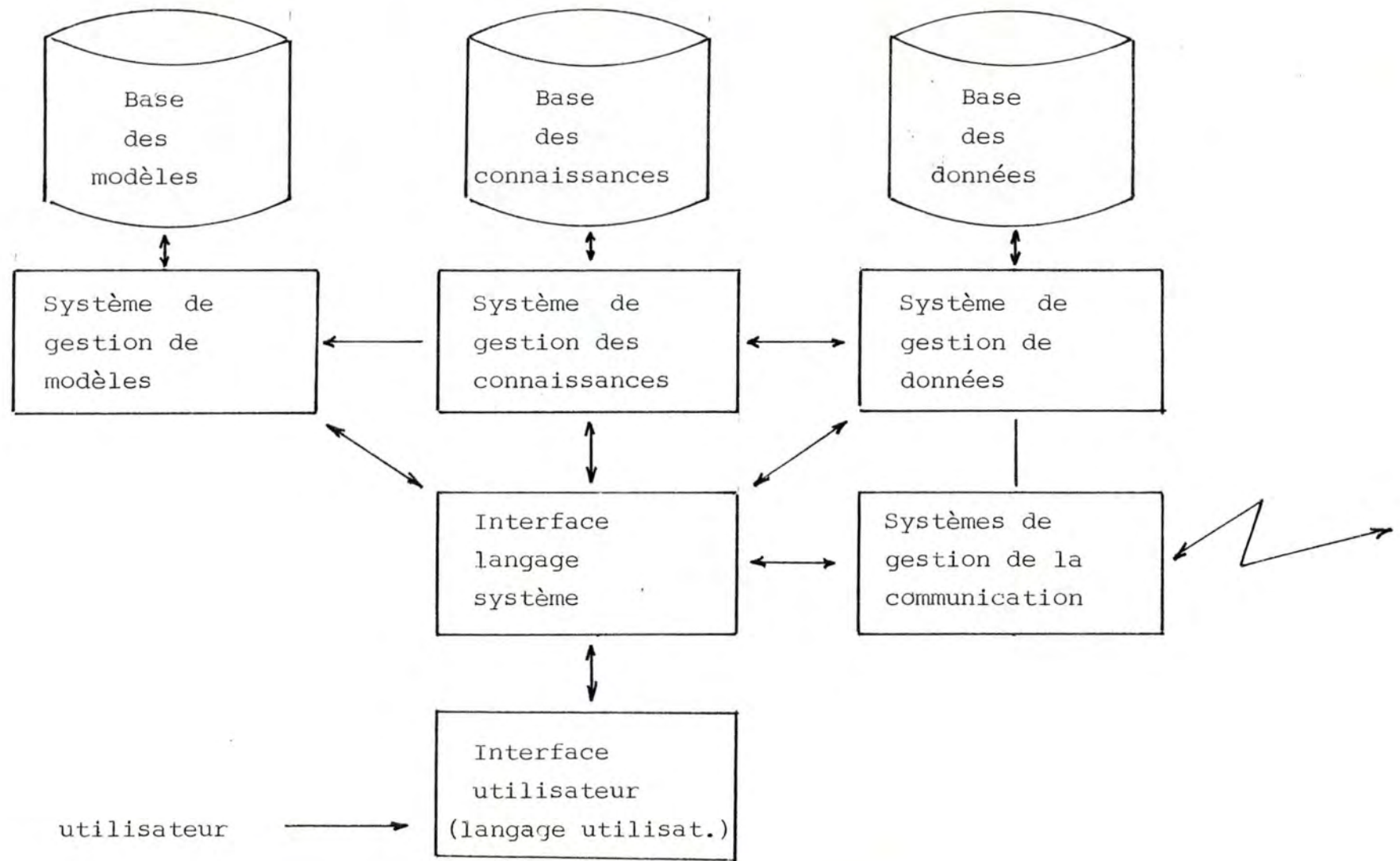


Figure 5.

- pour le système lui-même: le contenu de la base de connaissances peut être "interprété" dans un processus de génération.

Plusieurs méthodes de représentation de la connaissance peuvent être utilisées:

- La base de connaissances peut être organisée sous la forme d'un modèle de structuration des informations (modèle E/A, modèle binaire, modèle relationnel,...)
- La connaissance peut être représentée selon les méthodes développées dans le domaine de l'intelligence artificielle: calcul des prédicats et systèmes de production.

Il peut même être utile de combiner plusieurs de ces méthodes de représentation dans un même système de connaissances (21).

II.2.2. LE SOUS-SYSTEME DES DONNEES.

La fonction de ce sous-système est évidemment la gestion des données.

Les S.G.B.D. traditionnels doivent cependant être "améliorés" pour rencontrer certaines exigences S.I.A.D.

Ils devraient permettre notamment:

- la restructuration dynamique de la B.D. (pour l'enregistrement de données résultant de l'utilisation du système)
- l'accès rapide et aisé aux données (langage de requête de haut niveau, possibilité de tri).
- la gestion de données historiques (couramment utilisées par le gestionnaire).

- la manipulation de séries chronologiques à l'aide de primitives simples.
- l'aggrégation de données (somme, moyenne, ...)
- la gestion de versions multiples de données (liées par exemple à différentes hypothèses de simulation).
- la production aisée de rapports et de graphiques.
- la possibilité de définir une structure de données en "termes utilisateurs".

Ils devraient également fournir des fonctions de "data-monitoring" (S.G.B.D. "actifs").

Un S.G.B.D. "actif" est défini par F.A. MANOLA (24, P.83) comme étant: " a D.B.M.S. capable of taking more or less independent action when user-specified conditions arise, rather than passively executing storage and retrieval requests".

II.2.3. LE SOUS-SYSTEME DES MODELES.

Etant un nouveau produit des "recherches-S.I.A.D.", ce sous-système est encore mal défini et peu développé.

Le sous-système des modèles pourrait, par exemple, fournir à l'utilisateur la possibilité d'assembler entre eux des "blocs" semblables à des sous-routines de manière à créer de nouveaux modèles supportant la quasi-totalité d'un problème.

En effet, généralement les S.I.A.D. offrent un ensemble de modèles séparés, chacun ayant trait à un aspect spécifique du problème (ex.: modèle de précision, modèle d'optimisation, modèle statistique,...).

La "communication" entre ces modèles reste, pour l'utilisateur, une procédure manuelle et intellectuelle.

Ce sous-système devrait donc pouvoir offrir: (22,p.19)

- la possibilité de cataloguer et de maintenir un grand nombre de modèles supportant différents problèmes de gestion.
- la possibilité de créer facilement et rapidement de nouveaux modèles.
- la possibilité de relier ces modèles entre eux.
- la possibilité de relier ces modèles aux données appropriées.
- la possibilité de gérer la base de modèles grâce a des fonctions "gestion de modèles" analogues aux fonctions "gestion de données":
 - . langage permettant de créer et modifier facilement des modèles.
 - . mécanismes de rangement, de classification, et d'accès aux modèles.

II.2.4. LE SOUS-SYSTEME DU LANGAGE.

L'Interface langage-système.

Cet interface est composé des compilateurs et interpréteurs qui traduisent les instructions et les demandes de l'utilisateur pour les transmettre aux autres composants du système.

L'Interface-utilisateur.

Ce composant contrôle le dialogue avec l'utilisateur. Il peut lui fournir une variété de moyens de communication: commandes, menus, graphiques,...,souris, "light-pen",...

III.2.5. LE SOUS-SYSTEME DE COMMUNICATION.

L'objectif de ce sous-système est de permettre le chargement de la B.D. du S.I.A.D. à partir de la B.D. du

système informatique de l'organisation (ou même à partir de données extérieures).

Le fait que le S.G.B.D. du S.I.A.D. peut être différent de celui du système informatique de l'organisation et que les données sont généralement structurées différemment, pose un réel problème d'implémentation. Des opérations de conversion de données relativement complexes peuvent être nécessaires.

CONCLUSION.

Les principaux composants logiciels des S.I.A.D. diffèrent peu des composants des systèmes traditionnels: seul le sous-système des Modèles apparaît comme un nouveau produit.

Notons cependant la nécessité de concevoir des S.G.B.D. "orientés S.I.A.D.".

L'application des techniques de l'Intelligence Artificielle est également spécifique aux S.I.A.D..

CHAPITRE III.

UN GENERATEUR DE TABLEAUX.

III.1. OBJECTIF DU SYSTEME.

Ce système a comme objectif de fournir aux gestionnaires la possibilité de résoudre certains de leurs problèmes de gestion grâce à l'utilisation TRANSPARENTE d'un logiciel de type "tableur".

III. 2. DESCRIPTION D'UNE GENERATION DE TABLEAU. (25,26)

La figure 5. est une représentation schématique de la procédure de génération de tableau: les tableaux présentant à l'utilisateur les informations correspondant à son problème de décision seront générées automatiquement à partir d'une B.D. de spécifications*, d'une B.D. d'exploitation et d'un dialogue homme-machine.

*: Par la suite, nous utiliserons également l'appellation "B.D. de connaissances", expression généralement utilisée dans la littérature relative aux S.I.A.D. (Knowledge Base). Nous noterons que la Base de Connaissances utilisée ici n'est pas du même type que celle employée dans les systèmes experts.

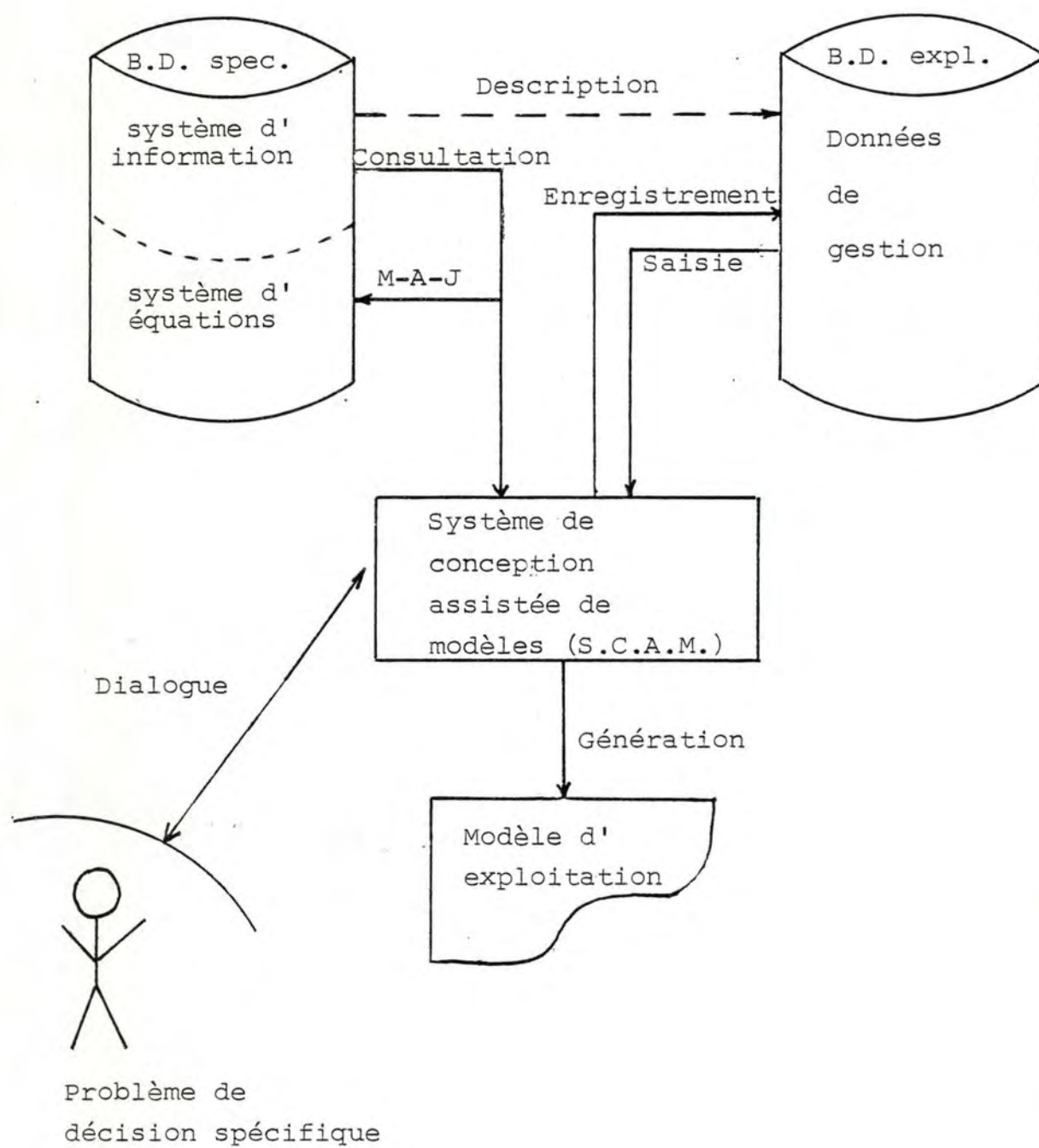


Figure 5.: Procédure de génération.

III.2.1. LES COMPOSANTS IMPLIQUES.

a) BASE DE DONNEES DE SPECIFICATIONS* .

LE SYSTEME D'INFORMATION contient la description des types d'information que le gestionnaire utilise à des fins de gestion.

LE SYSTEME D'EQUATIONS contient la description de modèles généraux.
Ces modèles sont constitués d'équations générales représentant les interrelations entre les différents types d'information de gestion utilisés par le gestionnaire.

b) BASE DE DONNEES D'EXPLOITATION.

Cette base contient les données de gestion utilisées par le gestionnaire.

c) MODELES D'EXPLOITATION.

Un modèle d'exploitation est un ensemble de données et d'équations permettant de calculer des données "résultat" à partir de ces données d'entrée. Un modèle d'exploitation est propre à un problème spécifique.

d) LE SYSTEME DE CONCEPTION ASSISTEE DE MODELE.

Ce système est le composant logiciel qui gère le dialogue avec l'utilisateur, transforme la spécification d'un problème en plans d'actions qu'il exécute.

* Le schéma conceptuel de cette B.D. sera décrit explicitement dans le chapitre I de la seconde partie.

III. 2. 2. RELATION ENTRE LES DIFFERENTS COMPOSANTS.

a) B.D. DE SPECIFICATIONS → B.D. D'EXPLOITATION;

Relation de DESCRIPTION: le système d'information de la B.D. de spécification décrit les données de la B.D. de spécification.

b) B.D. D'EXPLOITATION ↔ MODELES D'EXPLOITATION.

Via le S.C.A.M., grâce à la description du problème de décision de l'utilisateur:

- les données nécessaires à la génération du modèle d'exploitation correspondant à ce problème sont saisies automatiquement.
- les résultats provenant de l'exécution du modèle d'exploitation peuvent être enregistrés dans la B.D. d'exploitation.

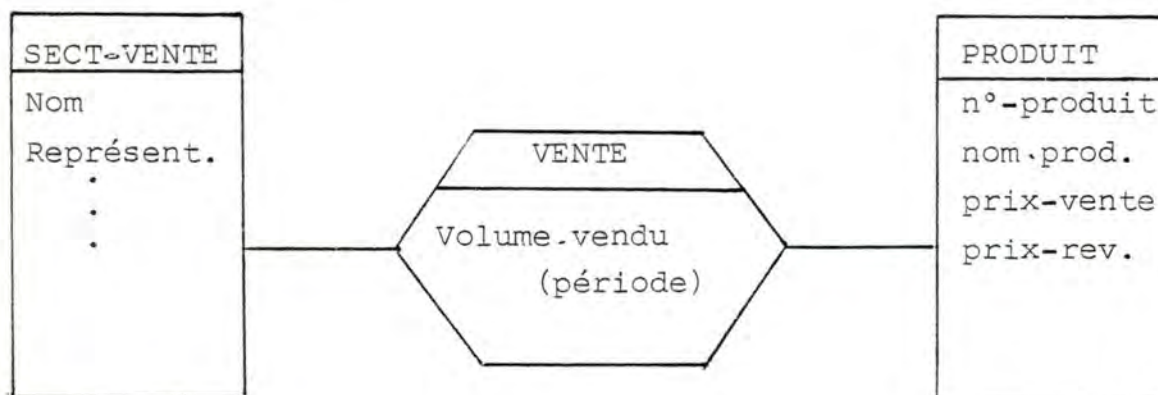
c) B.D. DE SPECIFICATIONS → MODELES D'EXPLOITATION.

Via le S.C.A.M., grâce à la description du problème de décision de l'utilisateur, les équations permettant de résoudre ce problème sont générées automatiquement.

III. 2.3. EXEMPLE DE GENERATION DE TABLEAU.

Soit le contenu de la B.D. de spécifications:

Système d'information:



Exemples de problème de décision.

a) Le gestionnaire envisage d'augmenter le prix de vente de ses produits. Il estime que cette modification aura des répercussions sur la vente et désire en évaluer la conséquence au niveau de l'entreprise. Il fournit donc au système la description de son problème de décision VIA UN LANGAGE DE DESCRIPTION DU PROBLEME.

ex.:

EFFET DE VARIATION DE

- . PRIX- VENTE (PRODUIT)
- . VOLUME-VENDU (SECT.-VENTE, PRODUIT)

SUR BENEFICE-TOTAL-VENTE

Le système consulte la B.D. de spécifications, s'aperçoit qu'il connaît toutes les informations nécessaires à la résolution du problème.

Il peut dialoguer avec l'utilisateur de manière à s'assurer que ce qu'il connaît correspond effectivement aux désirs de l'utilisateur.

exemple de dialogue:

SYST: désirez-vous visualiser le système d'équations et/ou d'information correspondant à votre problème ?

UTIL: Système d'équations.

SYST: [Impression des équations à l'écran]
(0) Désirez-vous ajouter des équations ?

UTIL: NON

Cet exemple représente une occurrence simple du méta-schéma relatif au système d'information. Il décrit la structure des informations concernant la vente des produits par les sections de vente.

On s'aperçoit donc que le volume vendu d'un produit est connu par section de vente et par période.

Système d'équations:

Le modèle général est défini par les équations suivantes:

$$\text{MARGE BENEFICIAIRE}(\text{PRODUIT}) = \\ \underline{\text{PRIX-VENTE}(\text{PRODUIT})} - \underline{\text{PRIX-REVIENT}(\text{PRODUIT})}$$

$$\text{BENEFICE}(\text{SECTION-VENTE}, \text{PRODUIT}, \text{PERIODE}) = \\ \underline{\text{VOLUME VENDU}(\text{SECT.-VENTE}, \text{PRODUIT}, \text{PERIODE})} \\ \times \text{MARGE BENEFICIAIRE}(\text{PRODUIT}).$$

$$\text{BENEFICE}(\text{PRODUIT}, \text{PERIODE}) = \\ \sum_j \text{BENEFICE}(\text{SECT.-VENTE}_j, \text{PRODUIT}, \text{PERIODE})$$

$$\text{BENEFICE-TOTAL-VENTE}(\text{PERIODE}) = \\ \sum_i \text{BENEFICE}(\text{PRODUIT}_i, \text{PERIODE})$$

Cet exemple représente une occurrence simple du méta-schéma relatif au système d'équation.

Il décrit le procédé de calcul du bénéfice total réalisé pendant une période, à partir des types d'informations décrits par le système d'information (ceux-ci sont soulignés dans les équations générales).

Le système propose alors le tableau suivant:

	A	B	C	D	E	F	G
1	N° PRODUIT	N° SECT. VENTE	PRIX VENTE	PRIX REVIENT	MARGE BENEFICI.	VOLUME	BENEFICE
2							
3	P1	SV1	INPUT	INPUT	C3 - D3	INPUT	E3 * F3
4	P1	SV2	INPUT	INPUT	C4 - D4	INPUT	E4 * F4
5	P1	SV3	INPUT	INPUT	C5 - D5	X	X
6							
7	P1	TOTAL					Σ
8							
9	P2	SV3	INPUT	INPUT	C9 - D9	INPUT	E9 * F9
10							
11	P2	TOTAL					Σ
12							
13
14
15
16
.							
.	Pn	SV1	INPUT	INPUT	X	X	X
.	Pn	SV3	INPUT	INPUT	X	X	X
.	Pn	SV4	INPUT	INPUT	X	X	X
.	Pn	SV5	INPUT	INPUT	X	X	X
m	Pn	TOTAL					Σ
m+1							
m+2	TOTAL						$G7 + G11 + \dots + Gm$

SYST: Ce tableau vous convient-il ?

UTIL: OUI.

SYST: Nouvelles valeurs des DONNEES DE SIMULATION ?
 UTIL: PRIX-VENTE(PRODUIT) \times 1,1.
 VOLUME- VENDU(PRODUIT) \times 0,95.
 SYST: PERIODE DE SIMULATION ?
 UTIL: 1985.
 SYST: Le modèle est généré sous le nom BENTOTVENT 1985.

b) Le dialogue aurait pu se dérouler de la manière suivante:
 (reprise à (0) du point a)).

SYST: [Impression des équations à l'écran]
 Désirez-vous ajouter des équations au modèle ?
 UTIL: OUI.
 SYST: AJOUTEZ LES EQUATIONS.
 UTIL: VOLUME- VENDU(PRODUIT, PERIODE)
 $= \sum_k \text{VOLUME- VENDU}(\text{SECT. - VENTE}_k, \text{PRODUIT}, \text{PERIODE})$

Le tableau résultant serait identique au tableau précédent
 au fait près qu'il calculerait en cases F7, F11, ..., F_n le sous-
 total correspondant à l'équation ajoutée.

SYST: Ce tableau vous convient-il ?
 UTIL: NON
 SYST: POURQUOI ?
 1. VARIABLE NON DESIREE
 2. VARIABLE MANQUANTE
 3. INVERSION LIGNES \rightarrow COLONNES
 :
 :
 UTIL: 2.
 SYST: Introduisez l'équation correspondant à la
 variable manquante !
 UTIL: VOLUME- VENDU(PERIODE) =
 $\sum_k \text{VOLUME VENDU}(\text{PRODUIT}_k, \text{PERIODE})$

Le tableau résultant contiendrait en case Fm+2 le total correspondant à l'équation ajoutée.

c) Soit une session ultérieure:

```

UTIL: EFFET DE VARIATION DE
      VOLUME-VENDU(PRODUIT)
      SUR BENEFICE-TOTAL-VENTE ?

      :
SYST: DESIREZ-VOUS AJOUTER DES EQUATIONS ?
UTIL: OUI
SYST: AJOUTEZ LES EQUATIONS
UTIL: BENEFICE(PRODUIT,PERIODE) =
      VOLUME-VENDU(PRODUIT,PERIODE) x
      MARGE BENEFICIAIRE(PRODUIT)

      :
      etc....

```

III.3. DESCRIPTION DE LA STRUCTURE GENERALE DU SYSTEME.

III.3.1. L'APPLICATION "EXPLOITATION et GESTION" DE LA B.D. D'EXPLOITATION.

L'exemple de génération décrit précédemment présente un système en fonctionnement dont la B.D. d'exploitation est chargée; les applications que l'utilisateur peut prendre en charge sont limitées au contenu de la B.D. d'exploitation.

Outre la possibilité de générer automatiquement des tableaux à partir de la spécification d'équations générales et des informations contenues dans la B.D. d'exploitation, l'utilisateur doit avoir la possibilité d'étendre lui-même le contenu de sa base d'exploitation en fonction des applications qu'il désire prendre en charge.

Cependant, les S.G.B.D. commercialisés sont complexes pour les utilisateurs non spécialistes: un utilisateur interprète le monde réel en termes "d'objets du monde réel" et d'actions sur ces objets.

Un S.G.B.D. ne travaille pas en termes d'objets ni d'actions du monde réel, mais en termes d'objets-S.G.B.D. et d'opérations - S.G.B.D..

Pour utiliser la Base de Données, l'utilisateur doit donc effectuer une transformation (mapping) de ses objets du monde réel et des actions à effectuer sur ceux-ci vers ceux de la B.D.

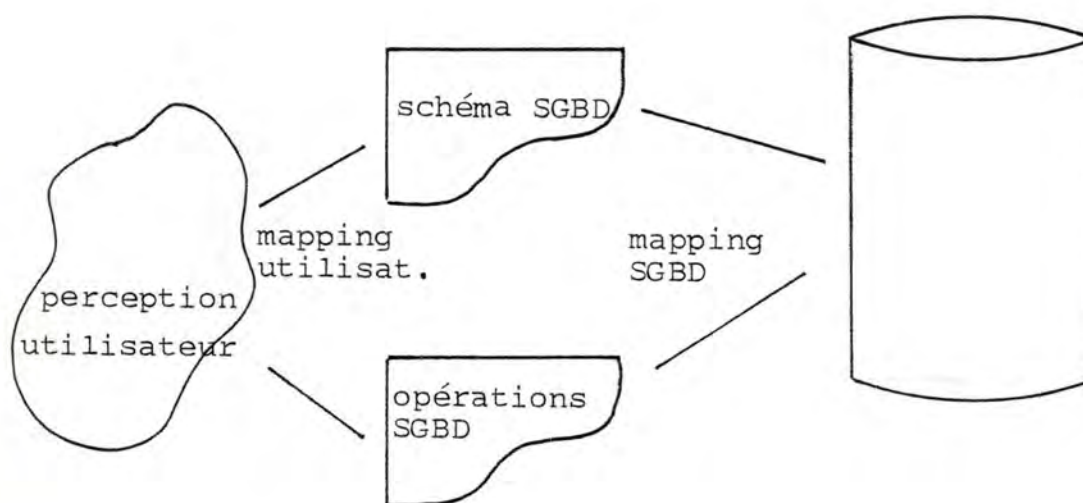
Le MODELE DES DONNEES détermine le type d'objets-S.G.B.D. et des opérations-S.G.B.D. que l'utilisateur doit employer pour interagir avec la B.D.

Les objets du monde réel, leurs relations et leurs caractéristiques doivent tout d'abord être décrits en termes des objets-S.G.B.D. légaux et des caractéristiques du modèle.

Ceci est fait par l'utilisation d'un langage de description de données (DDL) basé sur le modèle des données.

La collection des déclarations DDL qui définit le contenu de la base de donnée est appelé le SCHEMA de la B.D.

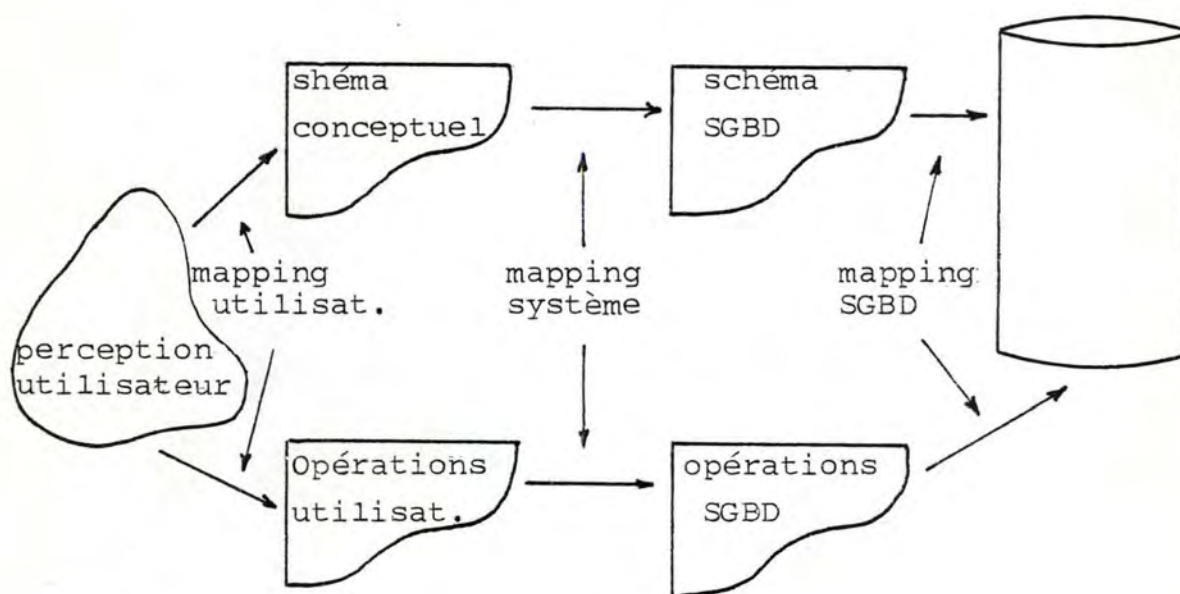
Les opérations doivent être soumises au S.G.B.D. par l'utilisation d'un langage de manipulation de données (DML) basé sur le modèle de données.



Puisque les modèles de données utilisés par les S.G.B.D. sont relativement complexes, il est nécessaire de fournir à l'utilisateur un modèle de données simple, plus facilement compréhensible et représentant mieux les "objets" du monde réel (MODELE CONCEPTUEL), ainsi que la possibilité d'opérer de manière simple sur la représentation de ces objets (OPERATIONS-UTILISATEUR).

Nous distinguerons donc deux fonctions du système, relatives à la gestion de la B.D. d'exploitation:

- 1) le mapping entre un schéma conceptuel et un schéma S.G.B.D.
- 2) le mapping entre "des opérations-utilisateur" et des "opérations-S.G.B.D.."



C'est par le biais du contenu du système d'informations de la B.D. de spécifications que le mapping entre les schémas s'effectuera.

La figure 5 représente la structure générale de l'application d' "exploitation et gestion de la B.D. d'exploitation".

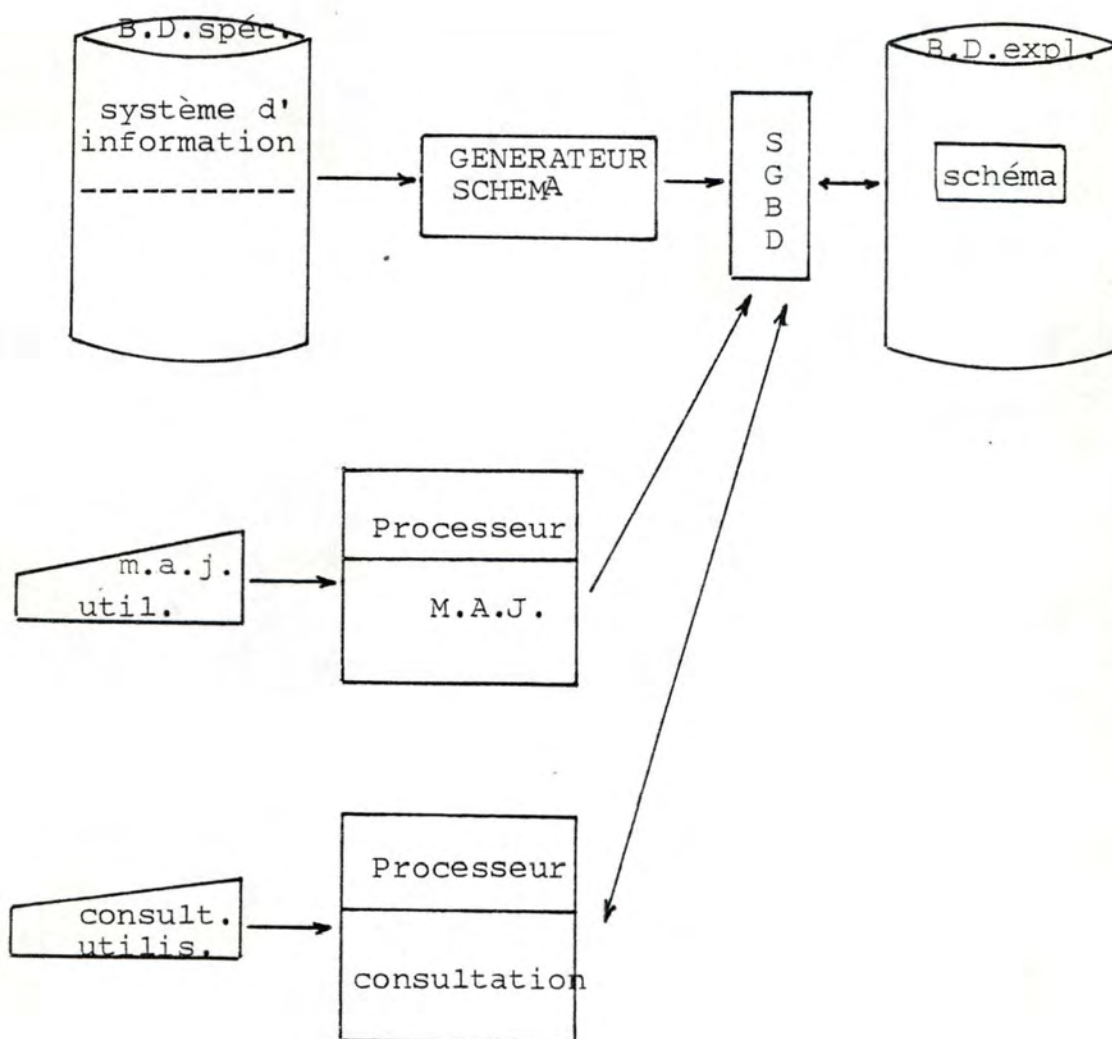


Figure 6.

III.3.2. L'APPLICATION "GESTION ET EXPLOITATION DE LA B.D. DE SPECIFICATIONS."

Il est également nécessaire de fournir à l'utilisateur la possibilité de consulter et mettre à jour le B.D. de connaissances.

Il disposera donc d'un langage de spécification d'équations générales, d'un langage de spécification de types d'information., ainsi que d'un langage d'interrogation.

Les outils de la B.D. de connaissance doivent pouvoir analyser les spécifications et mettre à jour la B.D., ainsi qu'effectuer certains contrôles (cohérence et complétude des modèles généraux).

La figure 7 représente la structure générale de l'application "exploitation et gestion de la B.D. se spécification".

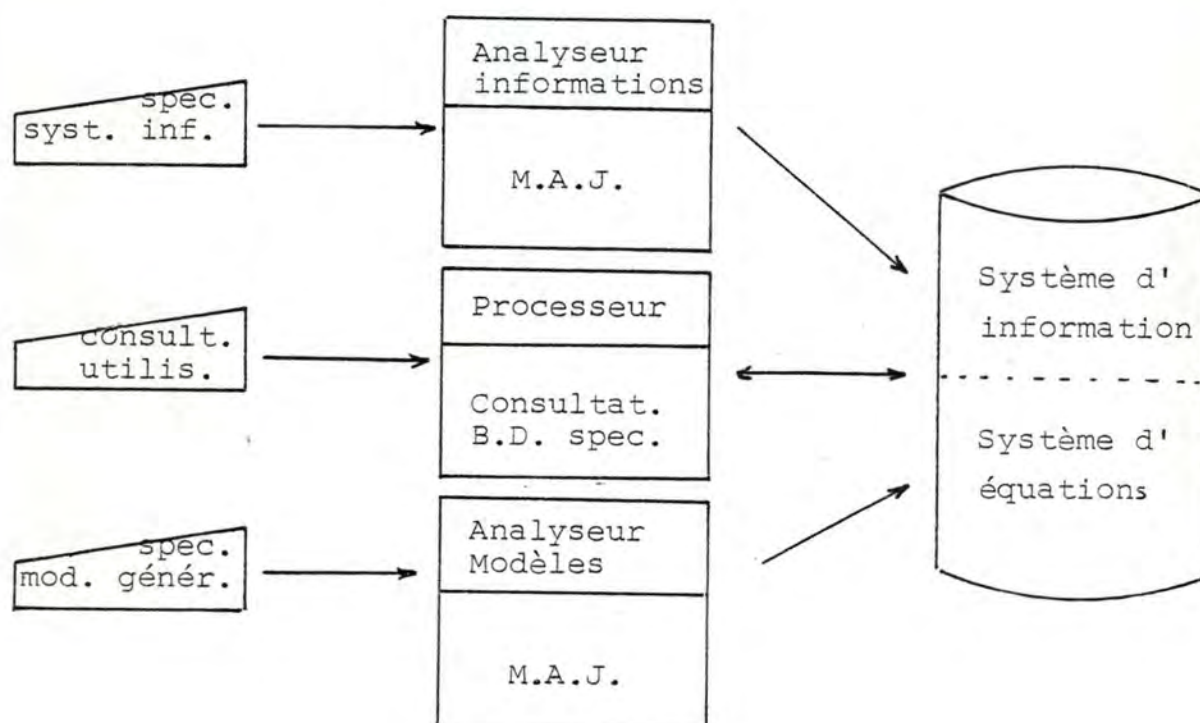


Figure 7.

III.3.3. L'APPLICATION "GENERATION-TABLEAU".

La génération de tableau (cfr. supra III.2) nécessite que le système puisse:

- concevoir une structure de tableau correspondant au problème de l'utilisateur.
- extraire de la B.D. d'exploitation les données qui interviennent comme données d'entrée dans le tableau.

Ceci pourrait se faire de la manière suivante: un ensemble d'équations formalisées (modèle) serait tout d'abord interprété pour générer

- un programme d'accès aux données (programme paramétré qui pourrait être utilisé à chaque génération correspondant à ce modèle).
- une description paramétrée du tableau.

Lors d'une demande de génération, le programme d'accès serait exécuté et la description du tableau serait exploitée pour générer le tableau spécifique au problème.

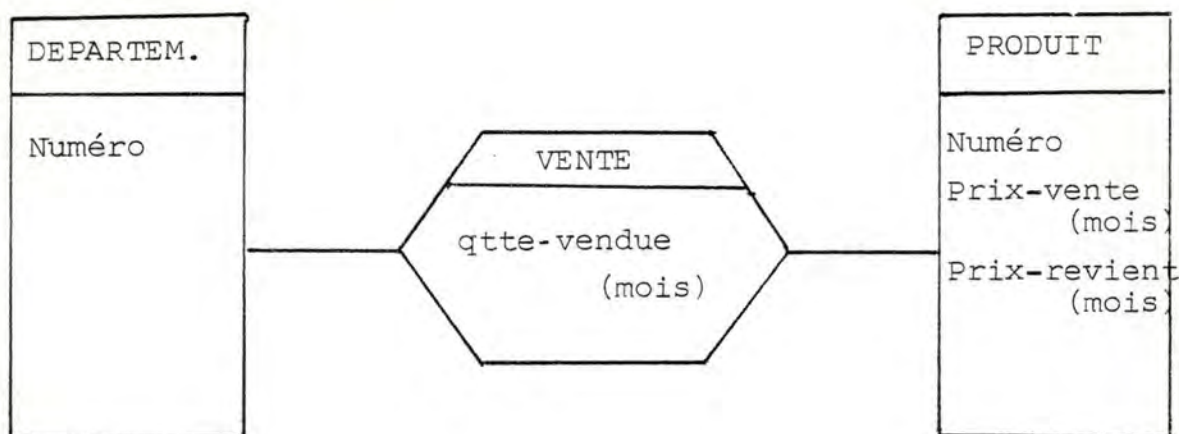
ILLUSTRATION*

Soit le système d'information:

*: Pour illustrer ce procédé, nous emploierons:

- un système d'information représentant explicitement les "caractéristiques temporelles" des attributs,
- le langage de formalisation d'équations que nous avons conçu pour ce système (chap. II dans la seconde partie).

Pour faciliter la compréhension, nous explicitons ces équations.



La quantité vendue de produit est connue par département et par mois.

Les prix de vente et de revient restent constants pendant le mois.

Soit le modèle composé des équations:

- (1) $BENEFICE (DEPARTEMENT/MOIS)$
 $= SOM (PRODUIT [VENTE] DEPARTEMENT ;$
 $BENEFICE DEPARTEMENT, PRODUIT/MOIS)) .$
 - (2) $BENEFICE (DEPARTEMENT, PRODUIT/MOIS)$
 $= QTTE-VENTUE (DEPART, PRODUIT, MOIS)$
 $\times MARGE-BENEF. (PRODUIT/MOIS) .$
 - (3) $MARGE-BENEF. (PRODUIT/MOIS)$
 $= PRIX-VENTE (PROD./MOIS) - PRIX-REVIENT (PROD./MOIS) .$
- (1) Le bénéfice réalisé par un département pendant un mois est EGAL à la somme, sur tous les produits vendus par ce département, des bénéfices réalisés sur ces produits par ce département pendant ce mois.
- (2) Le bénéfice réalisé par un département, sur un produit, pendant un mois est EGAL à la quantité vendue de ce produit par

ce département pendant le mois, MULTIPLIE par la marge bénéficiaire, pour ce mois, du produit.

- (3) La marge bénéficiaire d'un produit (pour le mois) est EGAL au prix-vente du produit (en vigueur pendant ce mois) MOINS le prix de revient (en vigueur pendant ce mois).

Modélisation de l'algorithme d'accès paramétré à générer.

```

N°-DEPARTEMENT → X:=
OCCUR. DE MOIS → Y:=

FOR DEP:=DEPARTEMENT(:NO-DEP=X)
FOR PRO:=PRODUIT VENTE DEP
    GET NO(PRO)
    FOR MO:=Y
        GET QTTE-VENDUE(DEP, PRO, MO)
        GET PRIX-VENTE(PRO, MO)
        GET PRIX-REVIENT(PRO, MO)
    ENDFOR
ENDFOR
ENDFOR

```

Tableau paramétré.

La description paramétrée pourrait définir le tableau suivant:

A	B	C	D	E	F	G	H
MOIS	N° DEP	N° PROD	Prix. VENTE	Prix. REVIENT	QTE. VENDUE	MARGE BENEF.	BENEFICE
						$C_i - D_i$	$E_i - F_i$
	TOTAL						Σ

Une demande de génération s'effectuera par la spécification d'un numéro de produit et d'une occurrence de mois.

Le programme d'accès sera exécuté (avec les arguments correspondants). Ce n'est qu'après la recherche des données que le tableau final pourra être généré.

La figure 8 représente la structure générale de l'application "génération-tableau".

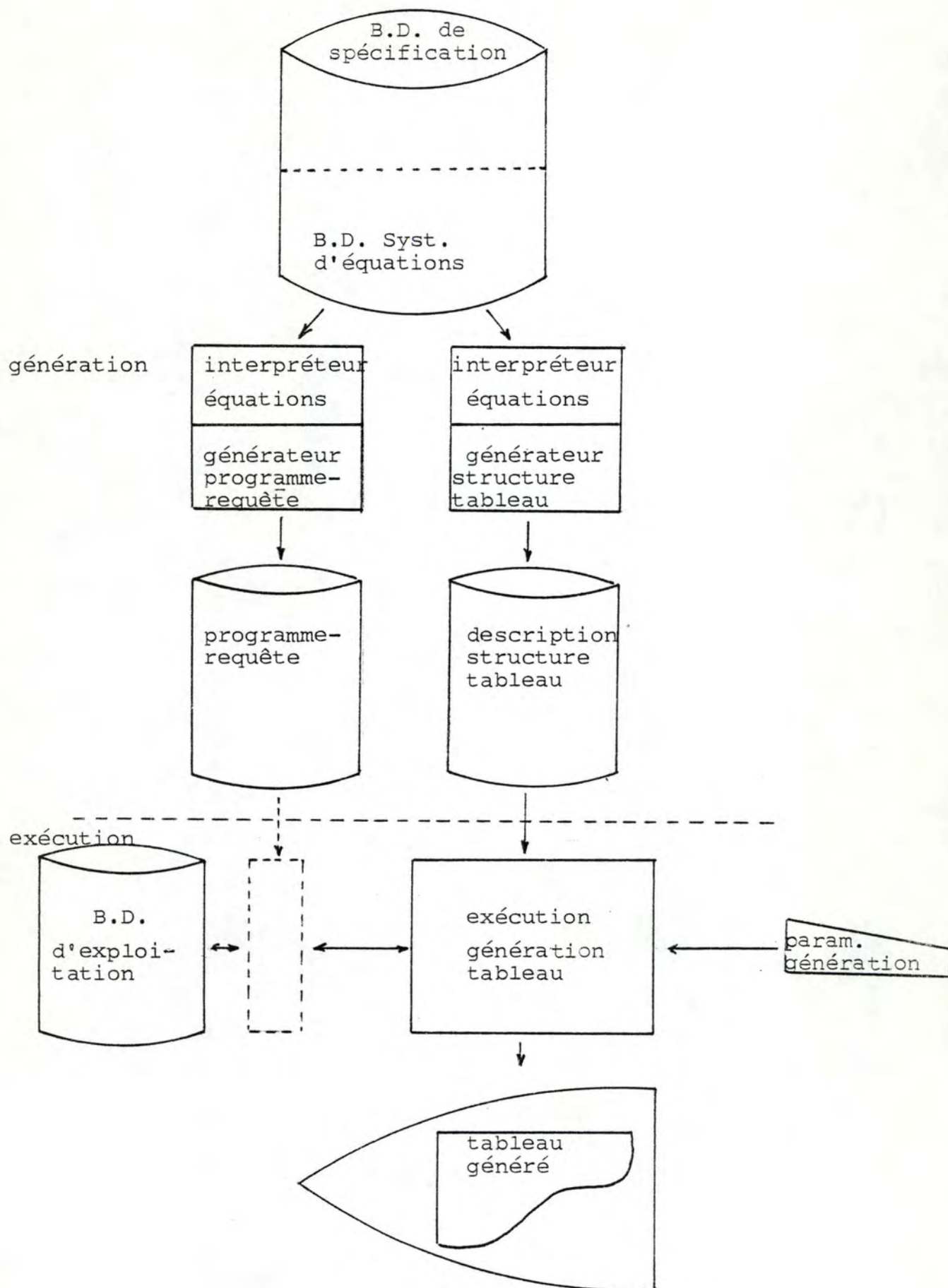


Figure 8.

CONCLUSIONS

Ce système est un "générateur S.I.A.D.". Cette qualification n'est pas due au fait qu'il génère des modèles d'exploitations, mais au fait qu'il doit pouvoir servir à construire différentes applications S.I.A.D. (cfr II.2.2. définition des "générateurs S.I.A.D.")

ex.: calcul de prix de revient, calcul de bénéfice, calcul de rendement,... dans des contextes différents: entreprise de production, entreprise commerciale,...

Il pourra donc être employé comme générateur pour tous les types de problèmes pouvant être supportés par un logiciel de type "tableur".

P A R T I E I I

ETUDE DU GENERATEUR

DE MODELES .

CHAPITRE I.

DESCRIPTION DU SCHEMA CONCEPTUEL DE LA BASE DE CONNAISSANCES DU GENERATEUR.

Il est nécessaire d'utiliser une méthode standard pour définir les types d'information susceptibles d'être enregistrés dans la base de connaissance.

La technique de modélisation choisie est le modèle E/A (28/29). Pour éviter les confusions dues au fait que ce modèle est lui-même décrit dans le méta-schéma, une terminologie différente sera utilisée.

TERMINOLOGIE ET CONCEPTS DE BASE.

Le schéma conceptuel du contenu de la base de connaissance sera décrit à partir des concepts suivants:

ELEMENT: tout type d'information au sujet duquel on désire enregistrer de l'information dans la méta-base.

ex.: les éléments "ENTITE", "ATTRIBUT", "MODELE".

RELATION: toute connection logique entre éléments.

ex.: la relation "CARACTERISE" entre les éléments "ENTITE" et "ATTRIBUT" cités précédemment.

PROPRIETE: toute caractéristique d'un élément ou d'une relation que l'on désire enregistrer dans la méta-base.

- ex.: - la propriété "DEFINITION" de l'élément "ENTITE".
 - la propriété "FORMAT" de l'élément "ATTRIBUT".

CONVENTIONS.

Nous définirons d'abord le schéma conceptuel du système d'information, ensuite le schéma conceptuel du système d'équations. Nous intégrerons ces deux sous-schémas en décrivant les relations composant l'interface entre ces deux sous-systèmes.

La description de chaque sous-schéma se fera selon le scénario suivant:

Chaque élément sera décrit par:

- sa définition
- ses propriétés
- les relations le connectant aux éléments déjà décrits précédemment.

Chaque relation sera décrite par:

- sa définition et par certaines précisions permettant de justifier les connectivités et les attributs de la relation
- une représentation graphique suivie, si nécessaire, de l'énoncé des contraintes non représentées.

Les propriétés sont décrites par leur définition et leur représentation dans la description des éléments ou relations qu'elles caractérisent.

La description graphique de chaque sous-schéma sera donnée en fin de description.

REMARQUES.

- Les représentations graphiques sont essentiellement utilisées pour faciliter la compréhension.

Excepté la mise en évidence des rôles joués par les éléments dans une relation et des connectivités de la relation pour les éléments qu'elle relie, elles n'ajoutent rien à la description.

- Les éléments et les associations sont univoquement définis par leur nom.

- Les noms des propriétés d'un élément (ou d'une relation) distinguent uniquement ces propriétés d'autres propriétés du même élément (ou de la même relation). Le nom permettant de distinguer univoquement une propriété parmi toutes les propriétés se compose du nom de la propriété et du nom de l'élément (ou relation) qu'elle caractérise.

- Cette remarque est également valable pour le nom du rôle joué par un élément au sein d'une relation.

INTRODUCTION.

Le schéma conceptuel de la base de connaissances a été élaboré en tenant compte des différents rôles qu'elle assume dans le système:

a) Pour l'utilisateur, la base de connaissances a une fonction essentiellement documentaire. Elle décrit les types d'informations qui sont à sa disposition dans la B.D. du système. Cette description doit lui permettre de formuler ses équations de manière correcte. Elle décrit également les modèles qu'il peut exploiter.

b) Le contenu du système d'information de la base de connaissances doit également pouvoir être exploité pour générer le schéma d'une B.D. historique* (les gestionnaires basent généralement leurs décisions sur des formations historiques).

c) Le système d'équations de la B.D. de connaissances doit pouvoir permettre au système de générer automatiquement le modèle correspondant à un problème de décision spécifique

*: Le problème de la localisation des informations dans le temps est exposé en annexe de ce travail.

I.1. DESCRIPTION DU SCHEMA CONCEPTUEL DU SYSTEME D'INFORMATION.

I.1.1. DESCRIPTION DES ELEMENTS.

I.1.1.1. DESCRIPTION DE L'ELEMENT "TYPE-ENTITE".

a) définition: une occurrence de l'élément "type-entité" correspond à l'existence dans le schéma conceptuel représentant l'activité de l'entreprise d'un type d'entité (T.E.).

Un type d'entité correspond à la spécification d'une classe de "choses" (concrètes ou abstraites) vérifiant une définition et considérées comme des unités fondamentales pour lesquelles on désire enregistrer de l'information.

ex.: une entreprise distingue le type d'entité "PRODUIT-FINI" correspondant à la définition suivante: "est considéré comme produit-fini tout produit commercialisable fabriqué dans l'entreprise".

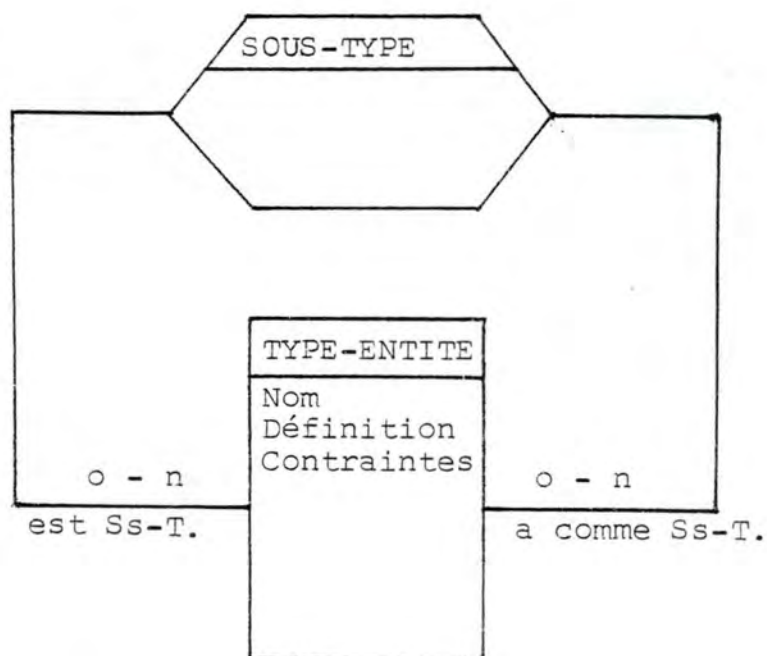
b) propriétés: les propriétés d'un type d'entité sont:

- le nom du type d'entité.
- la définition du type d'entité: description de la condition d'appartenance d'une "chose" à ce type.
- des contraintes d'existence: ces contraintes expriment des conditions (autres que celles spécifiées dans la définition du type) portant sur l'existence d'entités de ce type: contraintes liant la validité de l'existence d'une entité à d'autres éléments de la structure des données.

c) relation:. DESCRIPTION DE LA RELATION "SOUS-TYPE" ENTRE "TYPES-ENTITE".- définition:

Dans le contexte d'un système évolutif, il peut être utile de définir un type d'entité comme un sous-type (ou super-type) d'un autre type d'entité, afin:

- de permettre la classification d'une même occurrence de différentes manières.
- ou, de mettre en évidence la relation étroite entre un type d'entité défini à partir d'un type d'entité déjà existant.
- ex.: une entreprise commerciale vendant des produits qu'elle fabrique elle-même ou achetés chez un grossiste, peut définir les type d'entités "PRODUITS-FINIS" et "PRODUITS ACHETES" comme sous-types du type d'entité "PRODUITS COMMERCIALISES".

- représentation graphique:

I.1.1.2. DESCRIPTION DE L'ELEMENT "TYPE-ASSOCIATION".

a) définition: une occurrence de l'élément "type-association" correspond à l'existence, dans le schéma conceptuel représentant l'activité de l'entreprise, d'un type d'association (T.A.) entre types d'entité (T.E.).

Un type d'association correspond à la définition d'une classe de relations définies entre les occurrences des entités appartenant aux T.E. mis en correspondance par le T.A.

Le nombre de participants non nécessairement distincts à un T.A. est égal au degré du T.A. Il ne peut donc être inférieur à 2.

ex.: une entreprise distingue le T.A. "VENTE" mettant en correspondance les T.E. "PRODUIT-FINI" et "MAGASIN". Ce type d'association exprime le fait que les produits finis sont mis en vente dans des magasins.

b) propriétés: les propriétés d'un type d'association sont:

- le nom du type d'association;
- la définition du type d'association.
- le degré du T.A.
- des contraintes d'existence (autres que celles explicitées dans la définition):
 - . contraintes liant la validité de l'existence d'une association du type à l'existence d'autres éléments de la structure de données.
 - . contraintes d'exclusion.
 - . contraintes d'inclusion.

c) relation:

- . DESCRIPTION DE LA RELATION "PARTICIPE" ENTRE UN TYPE-ASSOCIATION ET DES TYPES-ENTITE.

- définition:

Cette relation met en correspondance le T.A. et les T.E. participant à ce type d'association.

Un T.A. pouvant être défini entre T.E. identiques, nous devons introduire un nouvel élément "ROLE" permettant de distinguer les différents rôles joués par un même T.E. au sein d'un T.A.

Un rôle donné est assuré par un T.E. dans le contexte d'un T.A.

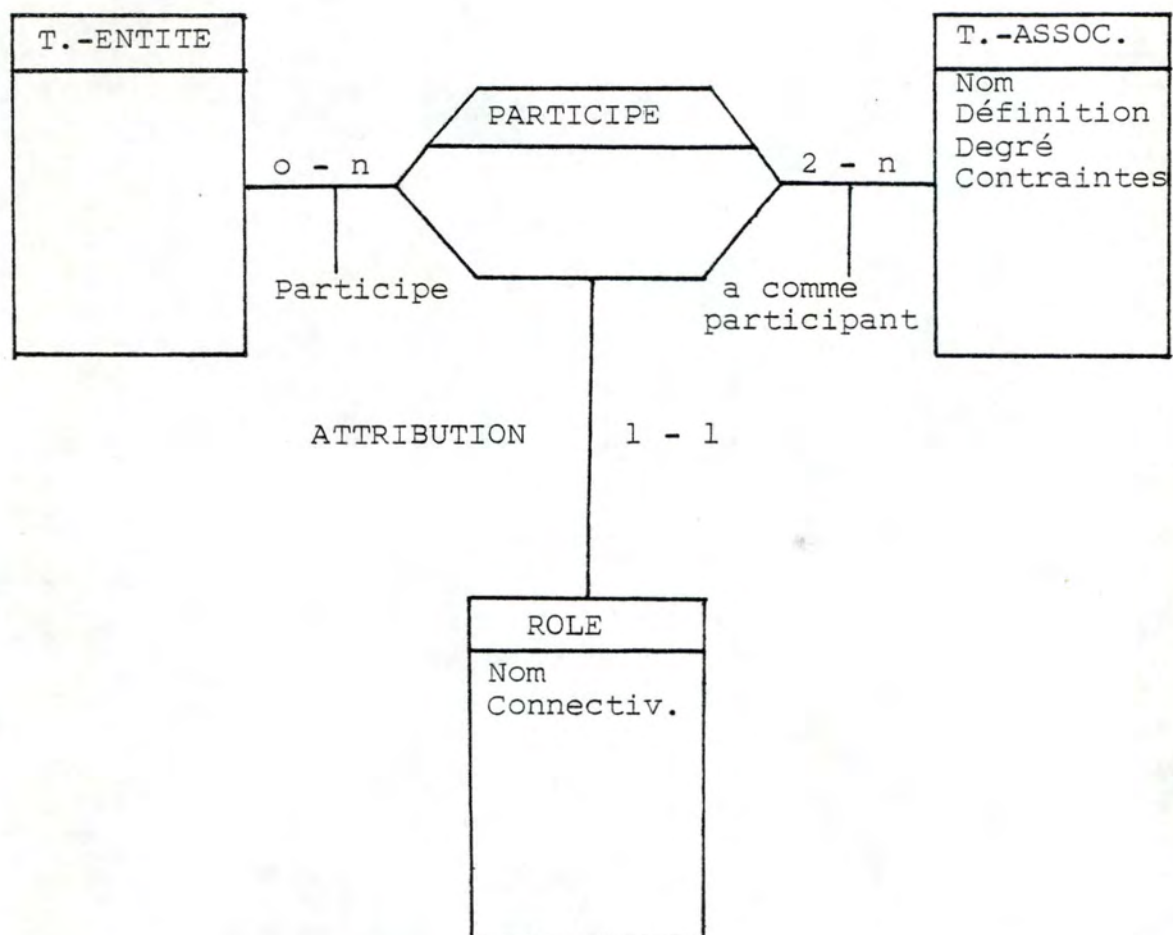
Il sera caractérisé par:

- un nom pouvant l'identifier.
- sa connectivité décrivant le nombre d'occurences du type d'association auxquelles une occurrence du type d'entité peut participer en jouant ce rôle.

Un T.E peut participer à plusieurs T.A. Il y assume nécessairement des rôles distincts.

Un T.A. met en correspondance des T.E. pouvant être identiques; mais y jouant des rôles différents. Le nombre de T.E. participant à un T.A. en y jouant un rôle donné est au moins égal à 2.

- représentation graphique:



I.1.1.3. DESCRIPTION DE L'ELEMENT "ATTRIBUT".

a) définition: une occurrence de l'élément "ATTRIBUT" correspond à l'existence dans le schéma conceptuel représentant l'activité de l'entreprise d'une caractéristique propre à un type d'entité ou à un type d'association.

ex.: - "PRIX-VENTE" attribut du T.E. "PRODUIT-FINI"
 - "QTTE-VENDUE-MENSUELLE" attribut du T.A.
 "VENTE" définie entre "PRODUIT-FINI" et
 "DEPARTEMENT".

b) propriétés: les propriétés d'un attribut sont:

- le nom de l'attribut.
- la définition de l'attribut.
- sa "décomposabilité".
- sa "cardinalité": un attribut peut être obligatoire ou optionnel. Il peut être également simple ou répétitif. Il est simple si, pour une occurrence d'un type d'entité ou d'association, il peut prendre au plus 1 valeur. Il est répétitif dans le cas contraire.

Nous regrouperons ces différentes caractéristiques en une seule propriété: la cardinalité de l'attribut: code permettant d'exprimer les contraintes portant sur le nombre minimal (obligatoire, optionnel) et maximal (simple, répétitif) de valeurs que peut prendre un attribut dans une occurrence de T.E. ou T.A. correspondant.

- les contraintes de format: code pouvant indiquer
 - . le type des caractères (alphanumériques, numériques, alphabétiques).
 - . le nombre de caractères.
- une unité de mesure: propriété permettant d'exprimer, si nécessaire, l'unité de mesure dans laquelle

la valeur de l'attribut sera exprimée.

ex.: PRIX du PRODUIT exprimé en FB.

QTTE-VENDUE-PAR MOIS de PRODUITS exprimée en FB.

NBRE-UNITES-VENDUES-PAR MOIS: sans unité de mesure.

- des contraintes de valeurs: contraintes portant sur l'ensemble des valeurs que peut prendre un attribut en fonction des valeurs prises par un autre attribut.
- sa "stabilité temporelle": code permettant de distinguer les attributs:
 - . "stables dans le temps": dont les valeurs ne varient pas pendant la durée de vie de l'entité ou de l'association qu'ils caractérisent (ex.: l'attribut LOCALISATION du T.E. BATIMENT)
 - . "instantanés": dont les valeurs sont susceptibles de varier pendant la durée de vie de l'entité ou de l'association qu'ils caractérisent l'attribut PRIX du T.E. PRODUIT.
 - . "à période d'évaluation" pour lesquels une attribution de valeur ne peut être faite que relativement à une période d'évaluation.
ex.: l'attribut QTTE-VENDUE-MENSUELLE du T.E. PRODUIT.

c) relations:

. DESCRIPTION DE LA RELATION "COMPOSITION" ENTRE ATTRIBUTS.

- définitions:

Un attribut peut être élémentaire ou décomposable.

Il est décomposable, si à une occurrence d'un T.E. ou d'un T.A., il fait correspondre un groupe de valeurs de types différents et peut être décomposable en autant d'attributs qu'il ya de types de valeurs dans ce groupe.

Il est élémentaire dans le cas contraire.

Un attribut décomposable peut être formé d'attributs eux-mêmes décomposables et/ou d'attributs élémentaires.

La relation "composition" entre attributs met en correspondance un attribut décomposable et les attributs le composant. Un attribut ayant été défini comme une caractéristique d'un T.E. ou d'un T.A., les attributs entrant dans la composition d'un attribut décomposable sont donc attributs du même T.E. ou T.A.

Afin d'éviter la redondance de spécifications, nous imposons que:

un attribut donné entre dans la composition d'au plus un attribut décomposable: la décomposition d'un attribut décomposable en attributs exclusivement élémentaires peut être obtenue via la décomposition de ses attributs décomposables en attributs élémentaires.

ex.: soit l'attribut décomposable "ADRESSE-PERSONNE" (A-P) du T.E. "PERSONNE" qui est composé de l'attribut élémentaire "RUE" (R) et de l'attribut décomposable "LOCALITE" (L).
L'attribut "LOCALITE" est lui-même composé de deux attributs élémentaires "N°-POSTAL" (N-P) et "VILLE" (V).

Cette contrainte signifie donc que les attri-

but "N-P" et "V" composent uniquement l'attribut "L" et que les attributs "L" et "R" composent l'attribut "A-P".

illustration:

structure de "A-P"
.....

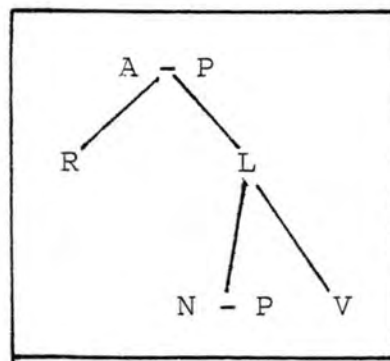
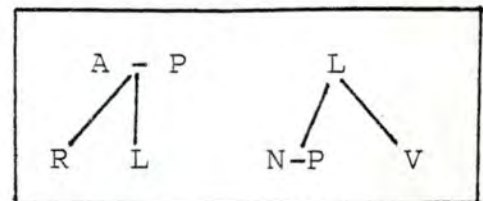


schéma d'occurences des
.....
spécifications.
.....



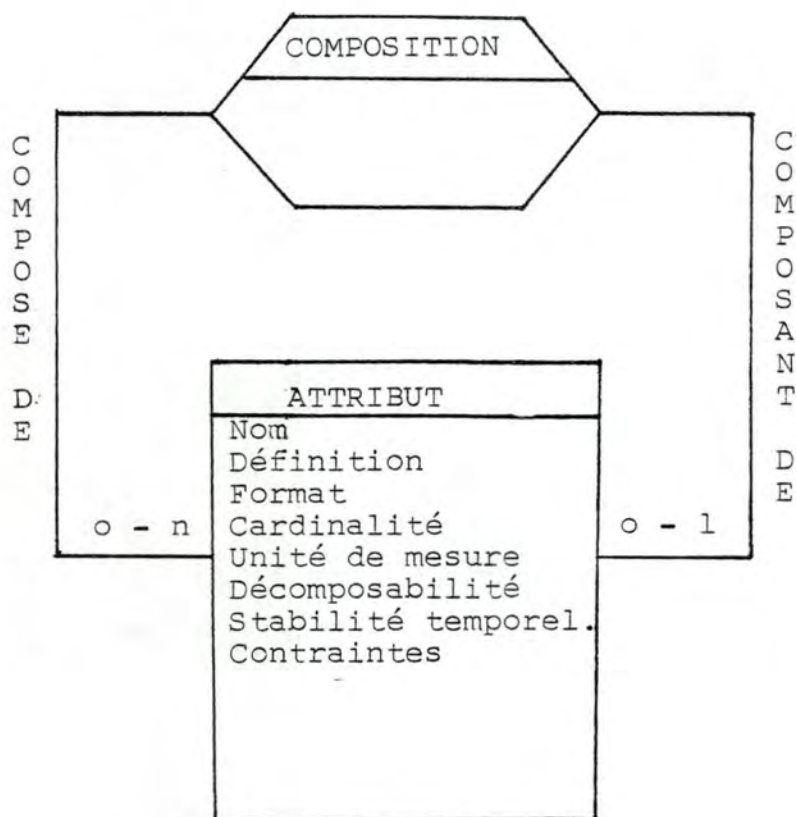
spécifications complètes
non redondantes privilég-
giées.

↓ : composé de
↑ : composant de

Nous privilégions cette méthode de spécifications car:

- elle semble plus naturelle.
- elle permet d'éviter la non-redondance par la formulation d'une règle simple (représentée par une contrainte de connectivité).
- la méthode de déduction de la spécification redondante non représentée est simple.

- représentation graphique:



- contraintes:

Seuls les attributs décomposables jouent le rôle "COMPOSE DE" au sein de la relation composition.

Les propriétés "format" et "unité de mesure" ont une valeur indéterminée pour les attributs décomposables (ces propriétés sont connues pour les attributs élémentaires correspondants).

La "cardinalité" d'un attribut jouant le rôle "COMPOSANT DE" dans la relation composition est identique à la cardinalité de l'attribut auquel il est relié par cette relation.

. DESCRIPTION DE LA RELATION "CARACTERISE-A" ENTRE UN TYPE D'ASSOCIATION ET DES ATTRIBUTS.

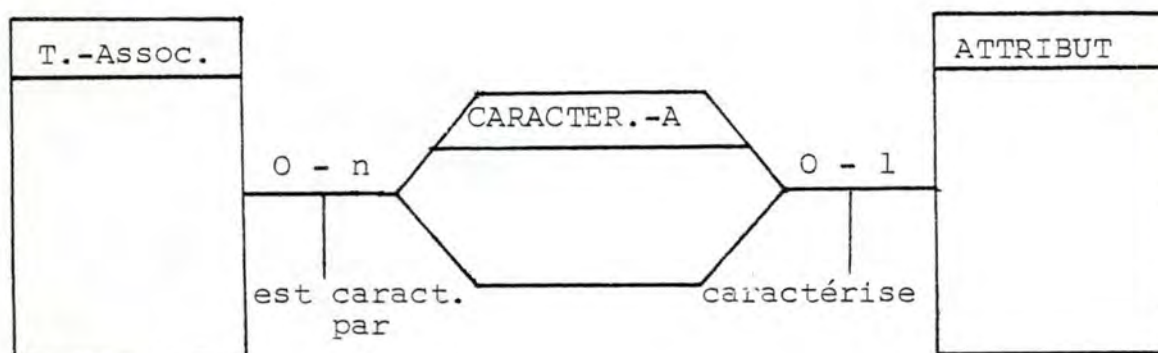
- définition:

Cette relation met en correspondance le type d'association et les attributs qui la caractérisent.

Un attribut peut être associé à 0 ou 1 type d'association: 0, car un attribut peut également être associé à un type d'entité.

Un type d'association peut posséder 0 ou plusieurs attributs.

- représentation graphique:



- contraintes d'intégrité:

La participation d'une occurrence d'"ATTRIBUT" dans une occurrence de "COMPOSITION", dans laquelle l'attribut assume le rôle "COMPOSANT DE" exclut sa participation à une occurrence de "CARACTERISE-A".

(la liste des attributs élémentaires peut être obtenue via la composition des relations "CARACTERISE-A" et "COMPOSITION".)

. DESCRIPTION DES RELATIONS ENTRE 'TYPE-ENTITE ET DES ATTRIBUTS.

Deux relations sont à distinguer: la relation de caractérisation, et la relation d'identification.

RELATION DE CARACTERISATION.

- définition:

Cette relation met en correspondance le type d'entité et les attributs qui la caractérisent.

Un type d'entité peut posséder plusieurs attributs. On admettra qu'il en possède au moins 1 (en effet, un type d'entité est décrit par ses attributs).

Un attribut caractérise 0 ou n type(s) d'entité:

- 0, car un attribut peut également caractériser un T.A.
- n, car nous avons introduit la relation "sous type" entre types d'entité.

Les T.E. sous-type d'un autre T.E. partagent les attributs du T.E. dont ils sont sous-types. Ils peuvent également avoir des attributs supplémentaires.

ex.: (cfr supra I.1.1.1.c), ex.)

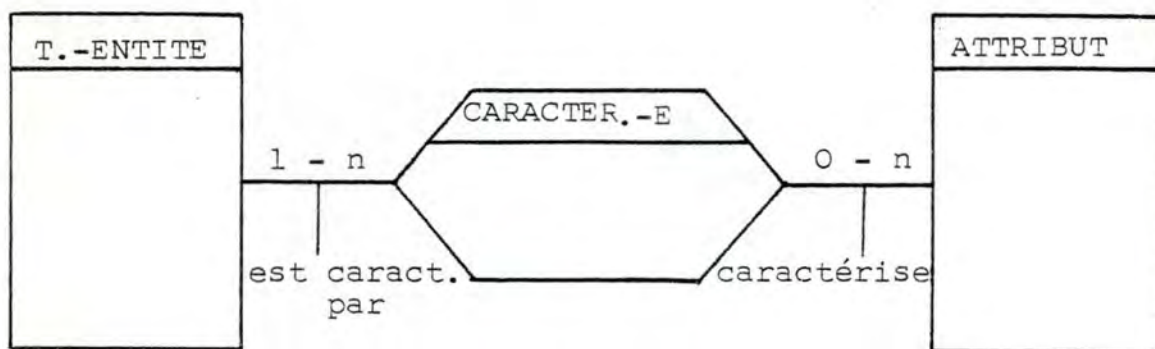
"PRODUIT COMMERCIALISE" peut être caractérisé par les attributs "N°-PRODUIT" et "PRIX-VENTE".

"PRODUIT-ACHETE" partage ces attributs et peut posséder l'attribut supplémentaire "PRIX-ACHAT".

"PRODUIT-FINI" partage ces attributs et peut posséder l'attribut supplémentaire "PRIX-REVIENT".

Pour tenir compte de ce fait, nous admettrons (et dans ce cas seulement) qu'un attribut puisse caractériser plusieurs T.E.

- représentation graphique:



- contraintes d'intégrité:

La participation d'une occurrence d'"ATTRIBUT" dans une occurrence de "COMPOSITION" dans laquelle il assume le rôle "COMPOSANT DE", exclut sa participation à une occurrence de "CARACTERISE-E" (la liste des attributs élémentaires d'un T.E. peut être obtenue via la composition des relations "CARACTERISE-E" et "COMPOSITION").

Une occurrence d'attribut ne jouant pas le rôle "composant de" participe nécessairement, soit à "CARACTERISE-E", soit à "CARACTERISE-A" (puisque'un attribut est défini comme caractéristique d'un T.E. ou d'un T.A.).

RELATION D'IDENTIFICATION.

Un attribut (ou groupe d'attributs) a un statut d'identifiant pour un T.E. si une valeur qui lui (leur) est attribuée permet de repérer univoquement une occurrence de ce type.

A un T.E. est associé au moins un identifiant.
Un T.E. peut posséder plusieurs identifiants distincts.

Deux cas sont à considérer:

cas 1) l'identifiant est un attribut

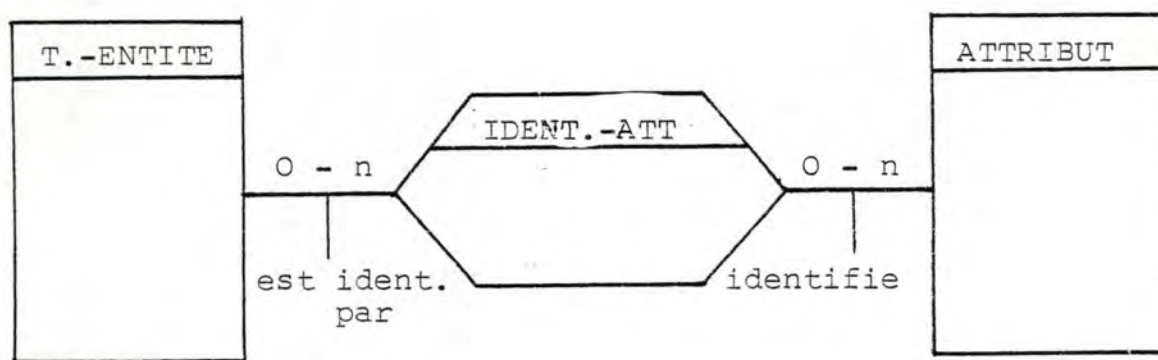
Description de la relation IDENT.-ATT entre
un T.E. et des attributs.
.....

Cette relation met en correspondance le T.E.
et les attributs l'identifiant.

Un T.E. peut être identifié par 0 ou n attributs. 0, car il peut être identifié par un groupe d'attributs.

Un attribut peut identifier 0 ou n T.E.; n, car nous avons introduit la relation "sous-type" entre types d'entité: un T.E. sous-type d'un autre T.E. partage l'identifiant du T.E. dont il est sous-type.

Représentation graphique.



Contraintes.

Les attributs participant à une occurrence de IDENT-ATT doivent être stables dans le temps.

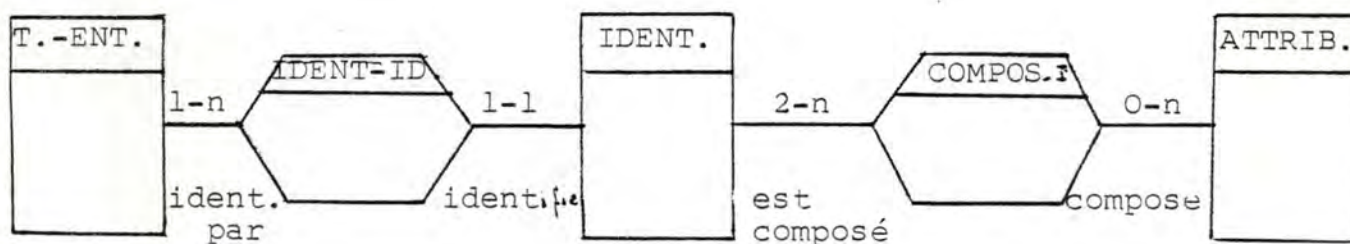
cas 2) l'identifiant est un groupe d'attributs.

De manière à distinguer les différents attributs d'un groupe d'attributs identifiant un T.E., nous introduirons un nouvel élément "IDENTIFICATEUR". Cet élément n'est introduit que comme mécanisme de groupement d'attributs différents identifiant un T.E. (il n'a qu'une propriété: un identifiant interne (géré par le système)).

Un type d'entité peut être identifié par 0 ou plusieurs identificateurs. Un identificateur est composé d'au moins deux attributs.

Un attribut entre dans la composition de 0 ou plusieurs identificateurs. Un identificateur identifie au moins un T.E. Il peut en identifier plusieurs (relation sous-type: même remarque que précédemment).

Représentation graphique.



La composition de ces deux relations met en correspondance le T.E. et les groupes d'attributs permettant de l'identifier.

Contraintes d'intégrité.

Une occurrence de "type-entité" doit participer, soit à une occurrence de "IDENT-ATT", soit à une occurrence de "IDENT-IDENT" (un T.E. possède toujours un identifiant).

I.1.1.4. DESCRIPTION DE L'ELEMENT "SYNONYME".

a) définition: une occurrence de l'élément "SYNONYME" correspond à la définition, pour les autres éléments de la structure de donnée; de noms pouvant les désigner univoquement (autres que celui repris dans la propriété "nom" des ces éléments).

b) propriété: la seule propriété de l'élément "SYNONYME" est son nom.

c) relations:

. DESCRIPTION DE LA RELATION "ENT-SYN" ENTRE TYPE D'ENTITE ET SYNONYME.

- définition:

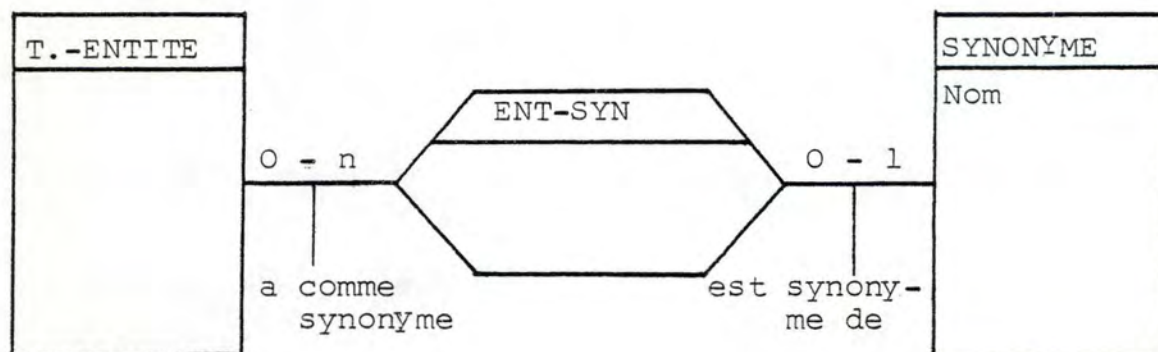
Cette relation met en correspondance un synonyme pouvant désigner le T.E. univoquement.

Un synonyme peut être associé à 0 ou 1 T.E.:

- à 0, car il peut être associé à un type d'association ou un attribut.
- à 1, car il doit identifier le T.E.

Un type d'entité peut posséder 0 ou plusieurs synonymes.

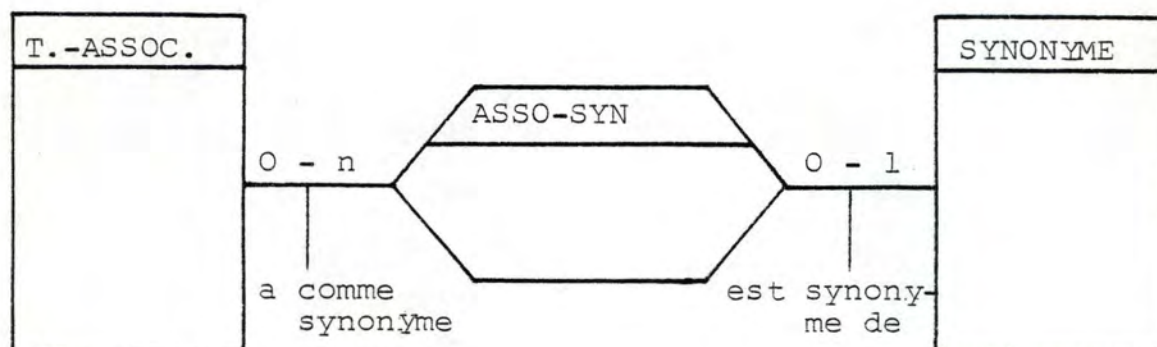
c) schéma:



. DESCRIPTION DE LA RELATION "ASSO-SYN" ENTRE TYPE-ASSOC ET SYNONYME.

La définition et les caractéristiques de cette relation étant similaires à celles décrites précédemment, seul le schéma sera donné.

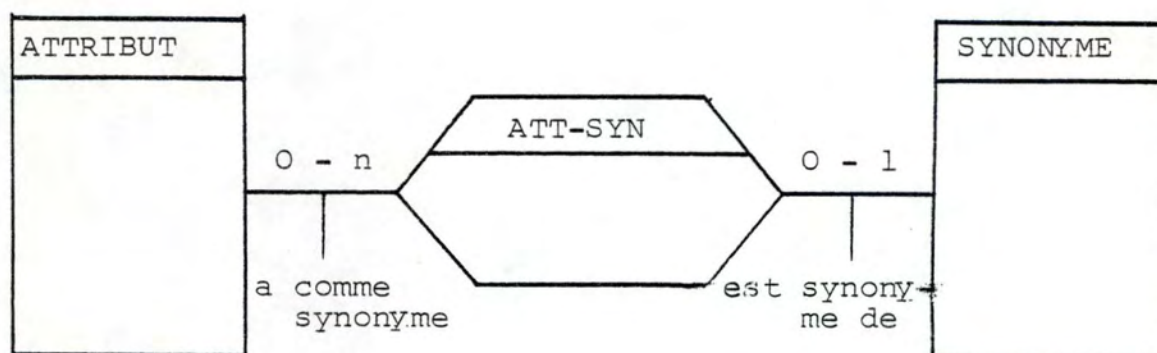
Schéma



. DESCRIPTION DE LA RELATION "ATT-SYN" ENTRE ATTRIBUTS ET SYNONYMES.

Même remarque que dans le point précédent.

Schéma:



I.1.1.5. DESCRIPTION DE L'ELEMENT "DOMAINE".

a) définition: un domaine est un ensemble de valeurs.

b) propriétés: les propriétés d'un domaine sont:

- le nom de l'identifiant: le nom d'un domaine peut parfois le définir implicitement (ex.: R).
- sa définition: propriété permettant d'exprimer sémantiquement l'ensemble des valeurs définissant le domaine.

ex.: domaine "AGE" = $\{x: x > 0 \text{ et } x < 100\}$

c) relations:

. RELATION DE "CONTRAINTE" ENTRE UN "DOMAINE" ET DES "ATTRIBUTS".

- définition:

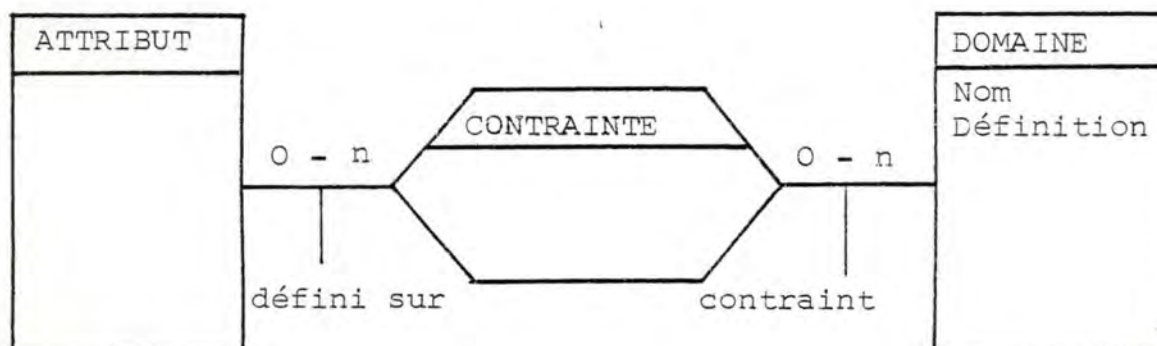
Cette relation exprime la contrainte portant sur des valeurs qu'un attribut peut prendre.

On admet qu'un attribut ne soit défini sur aucun domaine (ex.: attribut de type "texte").

Un même domaine peut être associé à des attributs différents.

ex.: le domaine "AGE" peut être associé aux attributs "AGE-EMPLOYE" et "AGE-OUVRIER".

- schéma:



I.1.1.6. DESCRIPTION DE L'ELEMENT "VALEUR".

a) définition: une valeur est une chaîne de caractères.

b) propriétés: elle n'a qu'une propriété: le symbole qui la représente et qui est la valeur elle-même.

c) relations:

. DSCRIPTION DES RELATIONS "DE CONTENU" ENTRE UN "DOMAINE" ET DES "VALEURS".

- définition:

Un domaine étant un ensemble de valeurs, il peut être nécessaire de définir un domaine par la liste des valeurs qui en font partie.

Nous distinguerons deux types de listes qui vont donner lieu à deux types de relation:

LA LISTE EXPLICITE.
.....

Cette relation met en correspondance le domaine et l'ensemble de toutes les valeurs le composant.

ex.: domaine "COULEUR" = ROUGE, VERT, JAUNE

LA LISTE IMPLICITE OU INTERVALLE.
.....

Cette relation met en correspondance le domaine et deux valeurs. Ces deux valeurs jouent le rôle de borne inférieure et borne supérieure du domaine. Les valeurs contenues entre ces deux bornes sont implicitement considérées comme faisant partie du domaine.

remarque: - les intervalles seront considérés comme des intervalles fermés.

- ce procédé nécessite l'existence d'une d'une relation d'ordre entre valeurs.

exemple: - domaine "AGE" = 0 .. 100

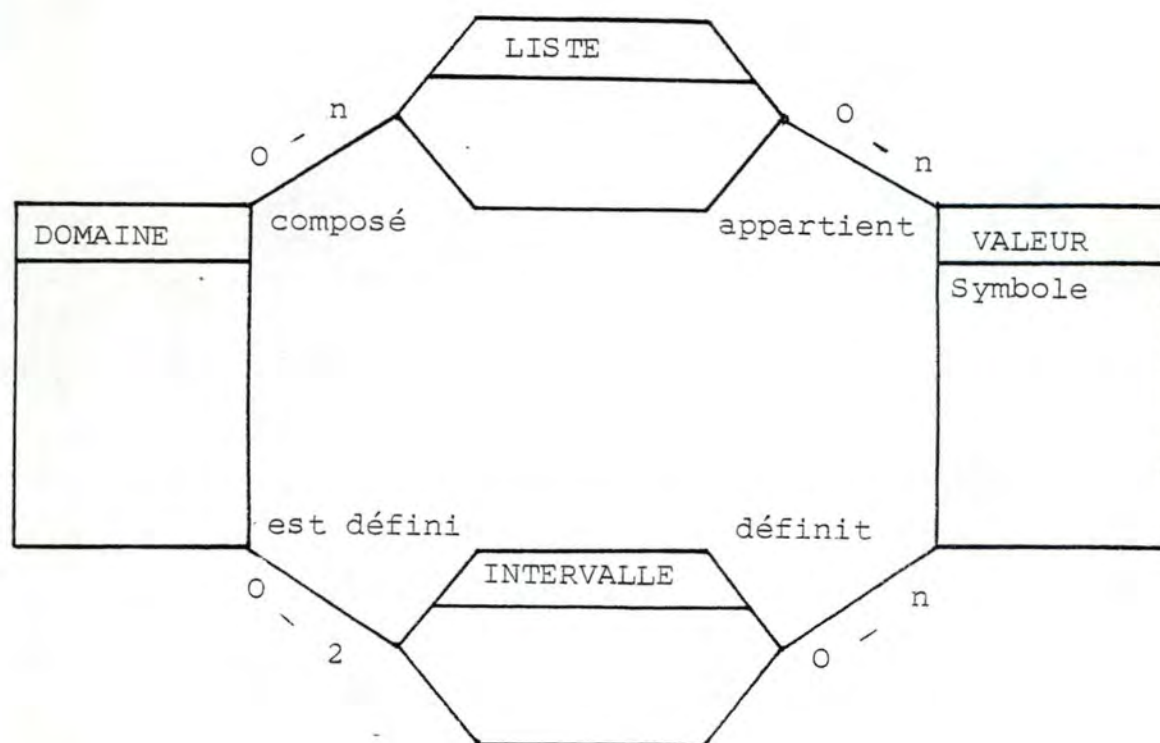
- domaine "JOURS OUVRABLES" = LUNDI ...
VENDREDI

On admet qu'un domaine soit défini par un autre procédé que la liste de ses valeurs (nom le définissant implicitement ou définition sémantique).

On admet qu'un domaine soit défini par un mélange d'intervalles et de listes de valeur (pas de contraintes d'exclusion).

Une valeur peut appartenir à plusieurs domaines.

-représentation_graphique:



- contraintes:

Les valeurs définissant un domaine doivent nécessairement vérifier le format décrit comme propriété des attributs définis sur ce domaine.

I.1.1.7. DESCRIPTION DE L'ELEMENT "TYPE-PERIODE".

REMARQUES PRELIMINAIRES.

La description des informations contenues dans la B.D. du S.I.A.D. pourrait se faire à partir des concepts de base décrits jusqu'ici. Il serait en effet possible de décrire les informations (tout en tenant compte de leurs caractéristiques temporelles) via le modèle E/A .

Exemple:

Considérons l'attribut "BENEFICE-DEPARTEMENT-MOIS" caractéristique du T.E. "DEPARTEMENT", pour lequel on désire enregistrer l'histoire des valeurs. ("Enregistrer l'histoire des valeurs" signifie: pour une occurrence de "département", chaque occurrence de "mois" doit lui faire correspondre une valeur définie sur le domaine "R")

Cet attribut pourrait être décrit, à partir des concepts de base du modèle E/A, comme étant un attribut décomposable et répétitif. Il serait composé des attributs élémentaires "BENEFICE-DEPARTEMENT" et "MOIS-DU-BENEFICE" définis respectivement sur les domaines "R" et "MOIS". Son caractère répétitif signifie que pour une occurrence de "DEPARTEMENT", il peut prendre plusieurs valeurs de type "R-MOIS" (l'enregistrement de l'histoire des valeurs est donc possible).

Cependant, la nature des informations manipulées par les gestionnaires (historique, actuelle, prévisionnelle) et les différents types de relations pouvant exister entre le temps et ces informations nous ont poussé à mettre le concept de TEMPS en évidence.

Une description à partir des concepts décrits jusqu'ici serait un "maquillage" des caractéristiques temporelles de l'information.

L'élément "TYPE-PERIODE" est introduit exclusivement comme procédé de représentation du TEMPS.

Les différentes interrelations pouvant exister entre le temps et l'information étant discutées en annexe, la description qui suit sera concise.

a) définition: une occurrence de l'élément "TYPE-PERIODE" correspond à l'existence d'un type de période reconnu comme pertinent pour l'entreprise.

ex.: MOIS, JOUR, ANNEE.

b) propriétés: les propriétés d'un type de période sont:

- son nom.
- le format de représentation de l'occurrence d'une période de ce type.

Nous envisageons, à partir de la propriété "format", pouvoir définir le CALENDRIER régissant les relations existant entre occurrences de période.

Exemple:

- Le format d'une période de type "JOUR" pourrait être JJ/MM/AAAA, celui d'une période de type "MOIS": MM/AAAA et enfin, celui d'une période de type "AN": AAAA
- Le calendrier ainsi défini serait tel que:

les occurrences de période de type "JOUR" seraient représentées dans la B.D. du S.I.A.D. par l'attribution d'une valeur à chaque "champ".

La valeur attribuée à "JJ" correspondrait au numéro d'ordre du jour dans le mois (occurrence) défini par les valeurs "MM/AAAA".

La valeur attribuée à "MM" serait le numéro d'ordre du mois dans l'année définie par les valeurs de "AAAA".

La valeur attribuée à "AAAA" serait le numéro d'ordre (défini à partir d'une origine) de l'année.

ex.:

Occurrence d'une période de type "jour"

31/12/1982 , 25/6/1983

Occurrence d'une période de type "mois"

12/1982 , 6/1978

Occurrence d'une période de type "an"

1982 , 1978

Il est donc possible, en privilégiant cette propriété, de définir une STRUCTURE de calendrier (basé sur une logique d'emboîtement de périodes).

Le calendrier correspondant serait implicitement défini par le respect de cette contrainte de format.

Ce procédé semble acceptable car il permet de définir un calendrier que nous qualifierons de "naturel". Il faut cependant remarquer que, pour qu'il soit applicable, les différents types de période à considérer doivent être prédéfinis.

En effet, le rôle privilégié attribué à la propriété "format" nous obligerait, lors de l'ajout d'un nouveau type de période, à modifier le format (et donc la représentation des occurrences) des autres types de période.

Exemple.

Soient les types de période "jour", "mois" et "année" définis comme décrit précédemment.

Supposons que l'on désire ajouter le type de période "TRIMESTRE". Le statut privilégié de la propriété "format" nous contraint à imposer à une période de type "jour" le nouveau format "JJ/MM/TT/AAAA".

Le calendrier correspondant serait défini de la manière suivante:

une valeur attribuée à "JJ" correspondrait au n° du jour dans le mois défini par la valeur attribuée à "MM/TT/AAAA" (nouveau format du type de période "mois").

La valeur attribuée à "MM" correspondrait au n° du mois dans le trimestre défini par la valeur de "TT/AAAA" (format du nouveau type de période), et celle attribuée à "TT" au n° du trimestre dans l'année "AAAA" (format inchangé).

c) relations:. RELATION ENTRE "TYPE-PERIODE".

Cette relation met en correspondance des types de période différents. Elle peut donc être exploitée pour définir la structure d'un calendrier.

Elle n'est citée ici qu'à titre indicatif et ne sera pas modélisée puisque nous avons choisi de privilégier la propriété "format" pour définir un calendrier implicite.

Dans un système d'information où le "temps" est un concept primordial, il était nécessaire de rappeler que

la notion de "calendrier" est également importante.

La structure du calendrier privilégié aurait pu être défini à partir d'une relation entre types de période.

. DESCRIPTION DES RELATIONS EXISTANTS ENTRE UN ATTRIBUT
ET UN TYPE-PERIODE.

RELATION "PERIODE-EVALUATION".
.....

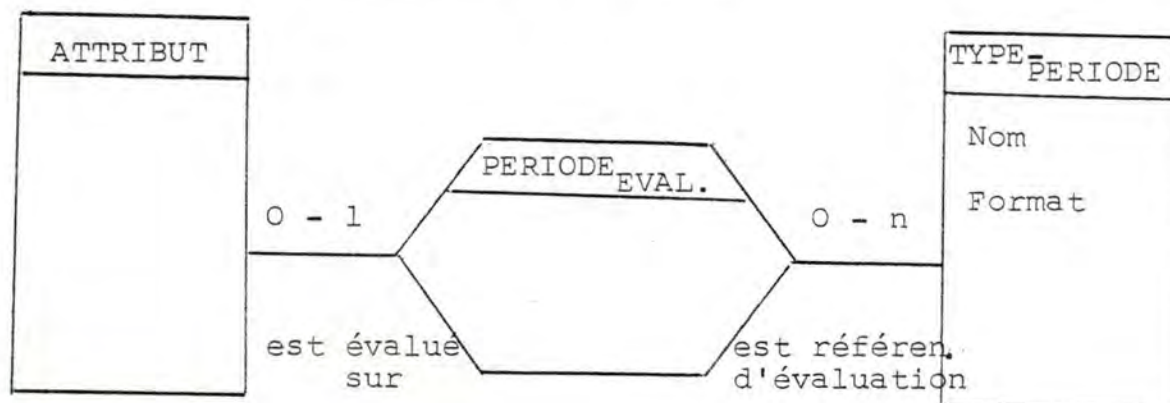
- définition: Cette relation met en correspondance l'attribut et le type de période sur lequel une valeur de cet attribut est évaluée.

Elle signale qu'une valeur de cet attribut pourra être localisée dans le temps par une occurrence de ce type de période (cette relation n'est pas uniquement l'expression d'une contrainte sur la durée de la période d'évaluation des valeurs de l'attributs).

A un attribut est associé un seul type de période d'évaluation (les attributs "QTTE-VENDUE-PAR-MOIS" et "QTTE-VENDUE-PAR-AN" sont des attributs différents).

Un même type de période peut être référentiel d'évaluation d'attributs différents.

- représentation graphique:



- contraintes:

Cette relation n'est définie que pour les attributs "à période d'évaluation".

RELATION "PERIODE-VALIDITE".
.....

- définition:

Cette relation met en correspondance l'attribut et le type de période (de durée maximale) sur lequel une valeur attribuée à cet attribut reste constante.

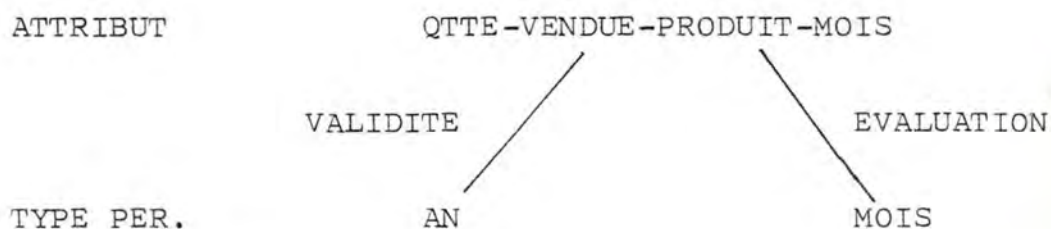
Cette relation est en fait l'expression d'une contrainte sur le comportement des valeurs de l'attribut dans le temps.

Elle n'est définie que sur des attributs non stables dans le temps (la période de validité des attributs "stables" est la durée de vie du S.I.).

Elle est définie pour tout attribut "instantané". Dans ce cas elle signifie qu'une valeur de cet attribut peut être localisée dans le temps par une occurrence du type de période qu'elle lui associe.

Elle est définie pour un attribut "à période d'évaluation" si le type de période que cette relation peut lui associer est différent du type de période qui lui est associé par la relation VALIDITE.

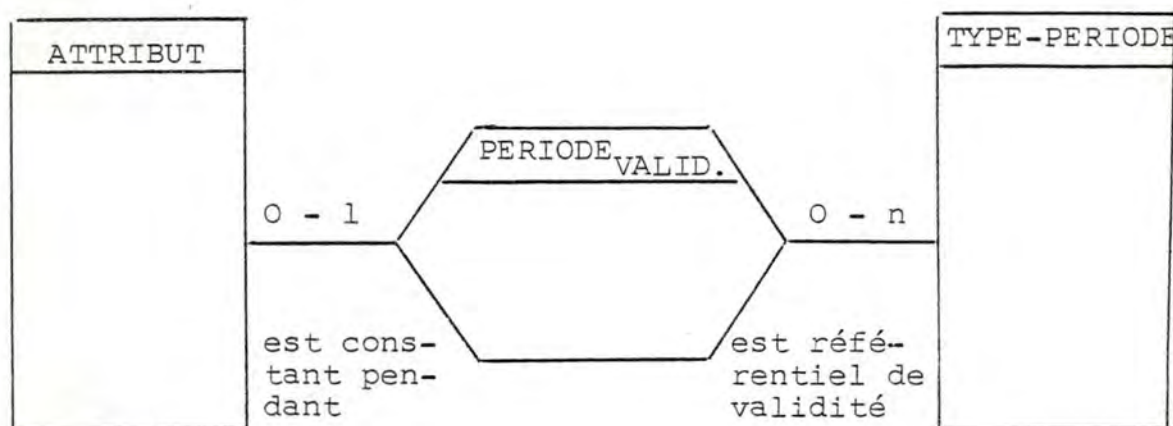
exemple :



Un même TYPE-PERIODE peut être référentiel de validité d'attributs différents.

A un attribut n'est associé qu'un seul TYPE-PERIODE de validité.

- représentation graphique:



- contraintes:

Une même occurrence de l'élément TYPE-PERIODE ne peut participer à la fois à une occurrence d'EVALUATION et de VALIDITE pour le même attribut .

(Il est évident qu'une valeur attribuée relativement à une période d'évaluation reste constante pendant cette période d'évaluation).

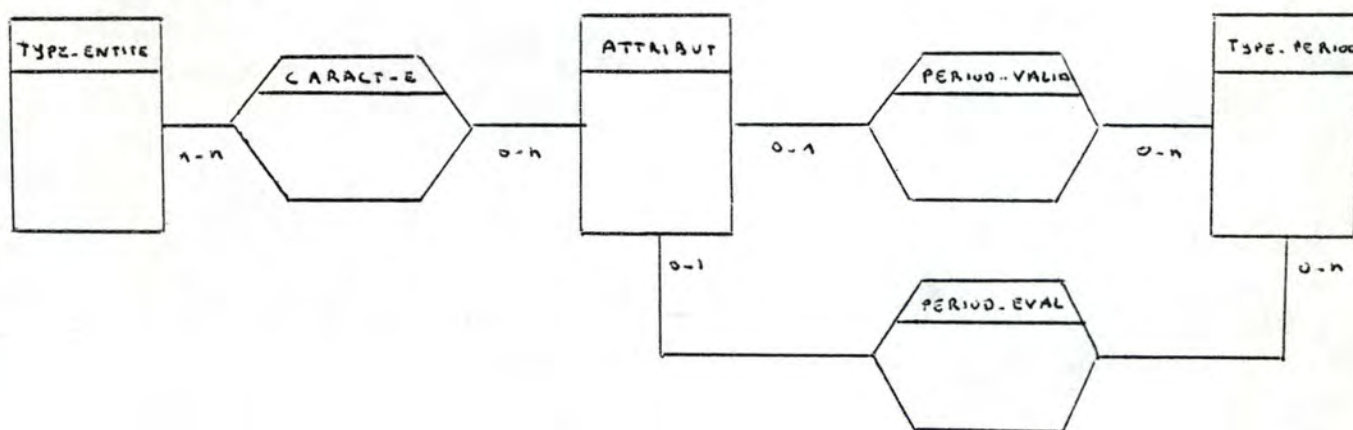
Un attribut "instantané" participe toujours à une occurrence de VALIDITE.

Seuls les attributs "à période d'évaluation" participent à la fois à VALIDITE et à EVALUATION.

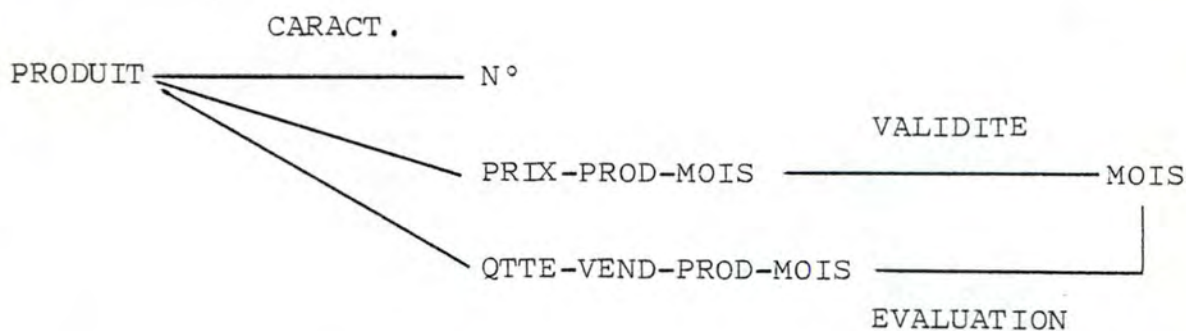
ILLUSTRATION

Soit l'occurrence suivante du méta-schéma représenté par la figure ci-dessous:

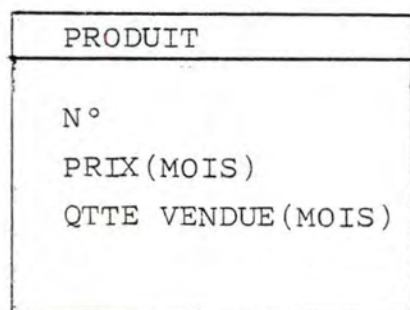
Méta-schéma.



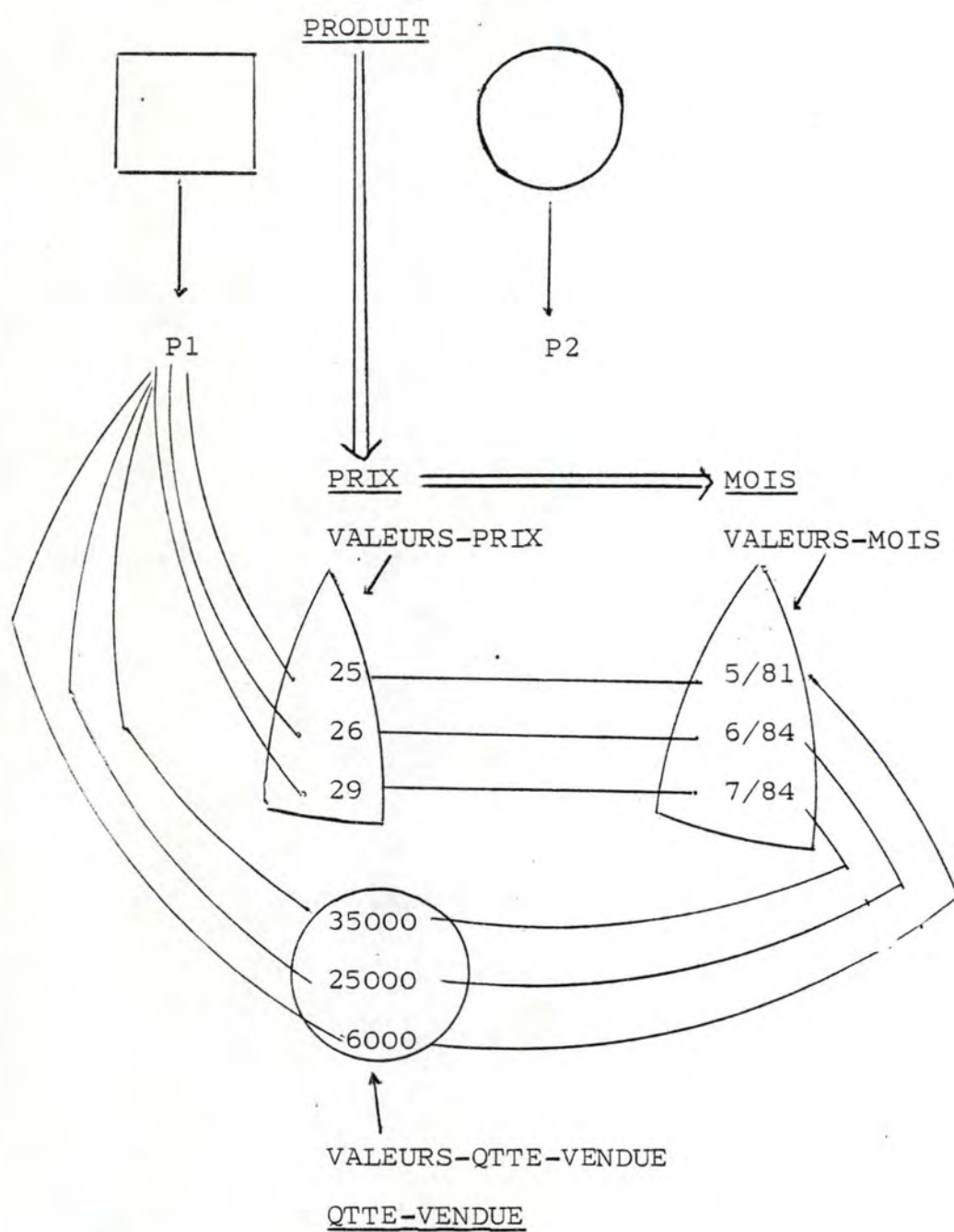
Occurence du méta-schéma.



pouvant être représentée graphiquement par:



Cette occurrence du méta-schéma pourrait par exemple elle-même décrire les occurrences suivantes:



Nous ne représenterons explicitement que les dépendances temporelles associées aux attributs: ce sont les seules qu'il est nécessaire de représenter pour permettre à l'utilisateur de formuler ses équations générales (cfr. annexe).

-ooo-

LE SCHEMA CONCEPTUEL INTEGRANT TOUS LES CONCEPTS
DECRIITS EST REPRESENTÉ PAR LA FIGURE 9.

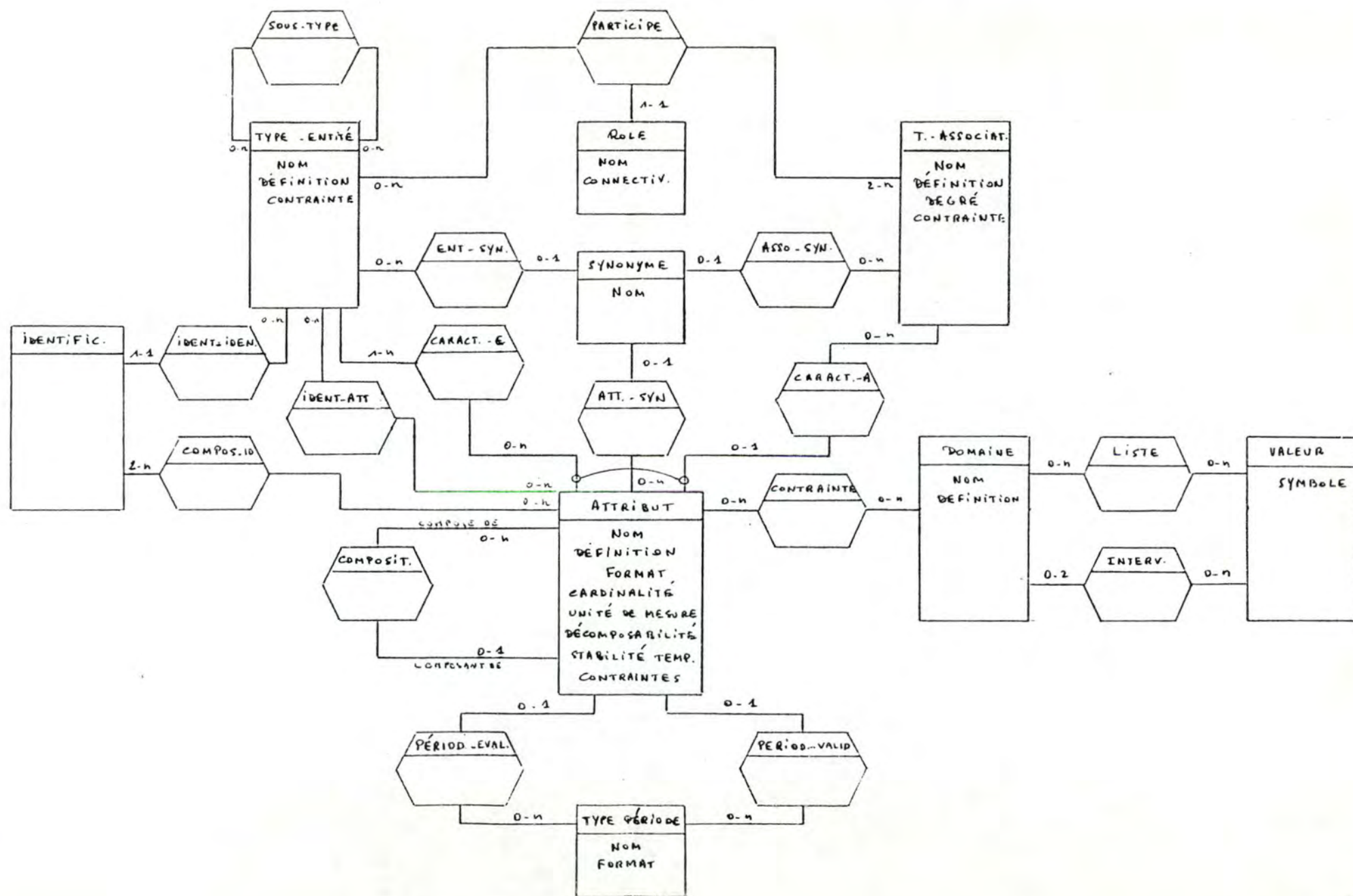


FIGURE 9

I.2. DESCRIPTION DU SCHEMA CONCEPTUEL DU SYSTEME D'EQUATIONS.

I.2.1. DEFINITIONS PRELIMINAIRES.

PROBLEME : un problème est une question à résoudre; il est lié à la volonté de connaître une (ou des) information(s) particulière(s).

exemples :

problème 1.: connaître le bénéfice réalisé par le département k pendant le mois j.

problème 2.: connaître le bénéfice réalisé par le département k pendant le mois i.

problème 3.: connaître le bénéfice réalisé sur un produit l pendant l'année m.

TYPE DE PROBLEME : un type de problème définit l'ensemble de tous les problèmes particuliers dont la résolution produit une information d'un certain type.

exemples:

Les problèmes 1. et 2. font partie de la même classe (type) de problèmes: "connaître le bénéfice mensuel d'un département".

Le problème 3. fait partie d'une autre classe de problèmes: "connaître le bénéfice annuel réalisé sur un produit".

Tout problème d'une classe de problèmes se résoudra par une méthode standard liée à la classe.

I.2.2. DESCRIPTION DES ELEMENTS.

I.2.2.1. DESCRIPTION DE L'ELEMENT "MODELE"

a) définition: un modèle est associé à un type de problème. La construction d'un modèle consiste à décrire le procédé de résolution STANDARD de tous les problèmes de ce type.

La spécification d'un modèle consiste en la spécification d'équations GENERALES décrivant ce procédé de résolution.

b) propriétés: un modèle sera décrit par - son nom
- sa définition.

I.2.2.2. DESCRIPTION DE L'ELEMENT "EQUATION".

a) définition: une équation est une expression arithmétique TAUTOLOGIQUE: elle définit une bijection entre un membre de gauche et un membre de droite.

b) propriétés: une équation n'a qu'une propriété, un numéro permettant de l'identifier (géré par le système).

c) relation:

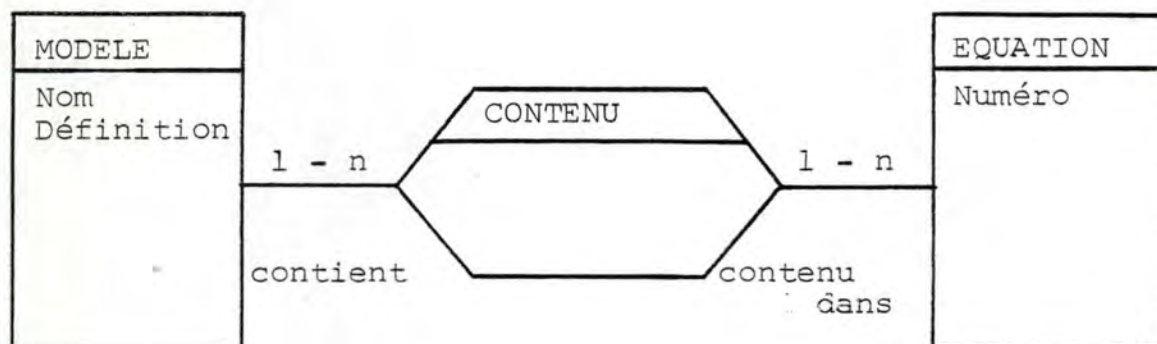
. DESCRIPTION DE LA RELATION "CONTENU" ENTRE DES EQUATIONS ET DES MODELES.

- définition:

Cette relation met en correspondance les modèles et les équations le composant. La décomposition d'un modèle en équations correspond à la décomposition d'un problème en sous-problèmes: la spécification des équations contenues dans le modèle est un moyen de description des actions à effectuer pour résoudre les sous-problèmes résultant de cette décomposition.

Un modèle contient au moins une équation. Une même équation peut être contenue dans des modèles différents.

- représentation graphique:



I.2.2.3. DESCRIPTION DE L'ELEMENT "VARIABLE".

a) définition: une variable est un type d'information calculé. Elle est donc membre de gauche d'une équation.

b) propriété: elle n'a qu'une propriété qui est son nom.

c) relation:

. DESCRIPTION DE LA RELATION "MEMBRE-GAUCHE" ENTRE UNE VARIABLE ET DES EQUATIONS.

- définition:

Cette relation met en correspondance la variable et les équations dont elle est membre de gauche. Une équation ne contient qu'un seul membre de gauche.

Une variable peut être membre de gauche d'équations différentes puisque le procédé de calcul d'un type d'information n'est pas unique.

exemple:

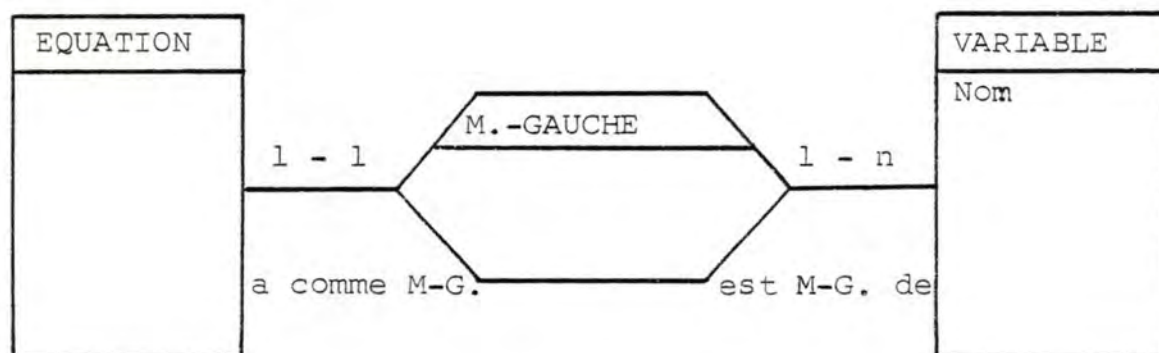
- (1) BENEFICE (PRODUIT/AN)
= SOM (MOIS [CALENDRIER] AN; BENEFICE (PRODUIT/
MOIS)).
- (2) BENEFICE (PRODUIT/AN)
= SOM (DEPARTEMENT [VENTE] PRODUIT; BENEFICE
(PRODUIT, DEPARTEMENT/AN)).

explication:

Le bénéfice réalisé sur un produit pendant une année =

- (1) somme sur tous les mois de l'année des bénéfices réalisés sur ce produit pendant un mois.
- (2) somme, sur tous les départements vendant le produit, des bénéfices réalisés par ces départements, sur ce produit pendant l'année.

- représentation graphique:



- contrainte.

Nous imposons que, dans le contexte d'un modèle, une variable est calculée de manière unique. Un modèle ne peut donc pas contenir deux équations différentes ayant même membre de gauche. Les deux équations décrites dans l'exemple ci-dessus ne peuvent donc appartenir à un même modèle (éviter les ambiguïtés)

. DESCRIPTION DE LA RELATION "RESULTAT" ENTRE UNE VARIABLE ET UN MODELE.

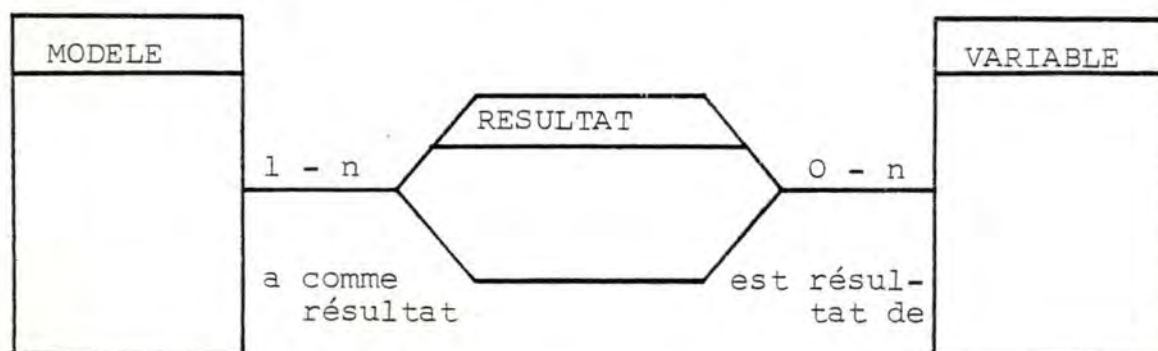
- définition:

Une occurrence de cette relation exprime que la variable est réellement un résultat du modèle, c'est-à-dire que le modèle a été conçu essentiellement pour décrire son procédé de calcul.

En effet, les informations calculées (variables) par les équations d'un modèle n'ont pas toutes le même "statut": certaines ne sont "que le résultat de calculs intermédiaires".

On admet qu'un modèle puisse fournir plusieurs résultats. Une même variable peut être résultat de modèles différents (puisque le procédé de calcul d'une variable n'est pas unique).

- représentation graphique:



- contrainte:

La participation d'une occurrence de VARIABLE dans une occurrence de RESULTAT implique sa participation dans une occurrence de MEMBRE-GAUCHE pour une équation du modèle en question.

- remarque:

Une variable participant à une occurrence de la relation RESULTAT sera, par la suite, parfois appelée "VARIABLE-RESULTAT".

I.2.2.4. DESCRIPTION DE L'ELEMENT "MEMBRE-DROITE".

a) définition: un membre de droite correspond à la description du procédé de calcul d'un membre de gauche d'une équation.

b) propriété: un membre de droite ne possède qu'une seule propriété: son expression.

c) relations:

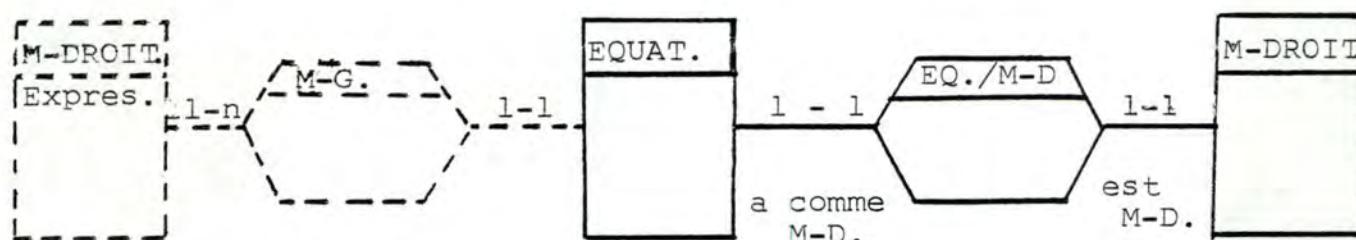
. DESCRIPTION DE LA RELATION "EQUATION/M-D" ENTRE UN MEMBRE DE DROITE ET UNE EQUATION.

- définition:

Cette relation met en correspondance l'équation et son membre de droite. Un membre de droite appartient à une seule équation.

L'élément "équation" a été introduit en fait pour établir une bijection entre une variable (membre de gauche) et un procédé de calcul (membre de droite). Le membre de droite est donc également un identifiant de l'équation.

- représentation graphique:



REMARQUE

description plus fine de l'élément
"MEMBRE DE DROITE".

On remarque que les membres de droite des équations générales décrites lors de la présentation du système (Part.I, ch.3) sont assez simples. En effet, ils ne contiennent que:

- des types d'informations DONT LA FORMULE DE CALCUL EST DONNEE PAR D'AUTRES EQUATIONS DU MODELE (donc membres de gauche d'équations du modèle): VARIABLES.
- des types d'informations NON-CALCULEES, DISPONIBLES DANS DANS LA B.D. DU S.I.A.D.

De manière à pouvoir, grâce à ce système, prendre en charge des problèmes "assez généraux", nous admettons qu'un membre de droite puisse également contenir:

- des types d'informations DONT LA FORMULE DE CALCUL EST DONNEE PAR LES EQUATIONS D'AUTRES MODELES (donc résultat d'un autre modèle):

- exemple schématique^{*} :

$A(X) = B(Y) + \text{résultat } F(X) \text{ du modèle "MNO"}$.

- des valeurs fixes.

- exemple:

$\text{POIDS-EN-KG}(\text{PRODUIT}) = \text{POIDS}(\text{PRODUIT}) / \underline{1000}$

en supposant que POIDS(PRODUIT) est disponible dans la B.D. du S.I.A.D. et y est exprimé en grammes.

- des types d'information NON CALCULES dont la valeur sera directement fournie à l'écran.

Ces distinctions doivent être mises en évidence dans le méta-schéma. En effet, elles sont nécessaires:

- à l'utilisateur qui doit connaître les types d'information qu'il manipule.
- au générateur de tableau qui, grâce à la consultation de la base de connaissance, doit pouvoir déduire:
 - a) quand générer des équations (seulement pour les types d'informations calculées).
quelles équations générer (quelles sont les équations générales décrivant les équations à générer pour résoudre le problème).
 - b) quand ne pas générer d'équations (pour les types d'informations non calculées).
comment accéder à l'information directement disponible (par demande au S.G.B.D. ou par demande à l'écran).

* Tout exemple schématique fait référence au langage de modélisation d'équations (décrit dans le chapitre II de cette partie). La notation A(X) désigne une caractéristique quelconque "A" d'un élément quelconque "X" (ex.: $\text{PRIX}(\text{PRODUIT})$).

Cette décomposition du membre de droite selon la nature des types d'informations qu'il contient nous conduit à définir de nouveaux éléments et nouvelles relations entre ces éléments et l'élément "MEMBRE-DROITE".

L'élément "VARIABLE" a déjà été défini.

. DESCRIPTION DE LA RELATION "COMPOSITION/V" ENTRE UNE VARIABLE ET UN "MEMBRE-DROITE".

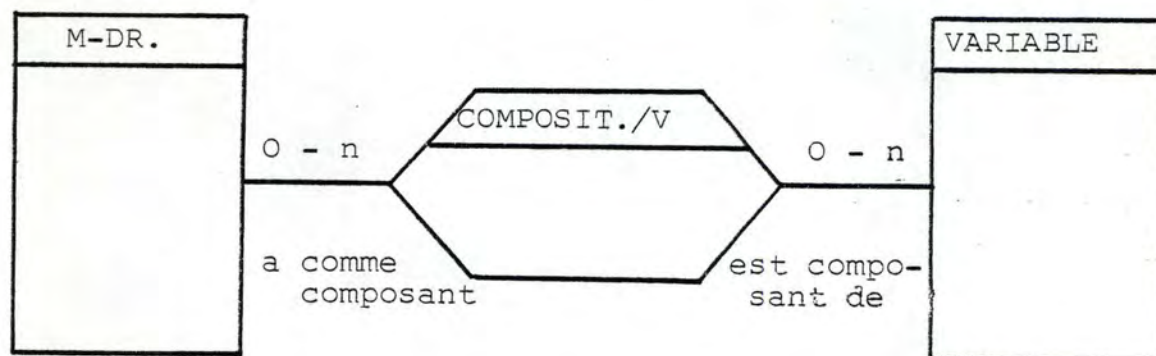
- définition:

Une occurrence de cette relation signifie que le type d'information entrant dans la composition du membre de droite est également fourni par une équation.

Un membre de droite peut contenir 0 ou plusieurs variables (zéro, puisque les types d'informations composant un membre de droite peuvent être de nature autre).

Une même variable peut intervenir dans 0 ou plusieurs membres de droite.

- représentation graphique:



- contraintes:

• Si une variable n'intervient dans aucun membre de droite, elle est nécessairement résultat d'un modèle (elle ne peut être "résultat de calculs intermédiaires" si elle n'est pas utilisée dans le processus de calcul d'autres variables):

La participation d'une occurrence de "VARIABLE" dans une occurrence de "MEMBRE-GAUCHE" pour une équation d'un modèle et sa non-participation à une occurrence de "COMPOSITION/V" pour une équation du même modèle implique sa participation à une occurrence de "RESULTAT" pour ce modèle.

• Un membre de droite constitue en fait un procédé de description (grâce à une formule générale de calcul) du type d'information (VARIABLE) qui lui est associé via la relation "MEMBRE-GAUCHE". Un membre de droite ne pourra donc faire référence au type d'information qu'il décrit:

(°) La participation d'une occurrence de "VARIABLE" dans une occurrence de "COMPOSITION/V" pour le membre de droite d'une équation exclut sa participation à une occurrence de "MEMBRE-GAUCHE" pour cette même équation.

Exemple schématique: On ne peut avoir d'équation générale de type:

$$\underline{A(X)} = B(X) + \underline{A(X)}$$

La contrainte exprimée par (°) n'est en fait pas suffisante: on ne peut avoir aucun "ensemble" d'équations générales tel que:

$$A(X) = B(X) + C(X)$$

$$C(X) = D(X) + E(Y)$$

$$E(Y) = F(Y) \times A(X)$$

puisque cet "ensemble" d'équations exprime que

$$\underline{A(X)} = B(X) + D(X) + F(Y) \neq \underline{A(X)}$$

ou encore

$$\underline{E(Y)} = F(Y) \neq (B(X) + D(X) + \underline{E(Y)})$$

I.2.2.5. DESCRIPTION DE L'ELEMENT "CONSTANTE".

a) définition: une constante est une chaîne de caractères numériques.

b) propriété: elle n'a qu'une propriété: la valeur qui la représente et qui est la constante elle-même.

c) relation:

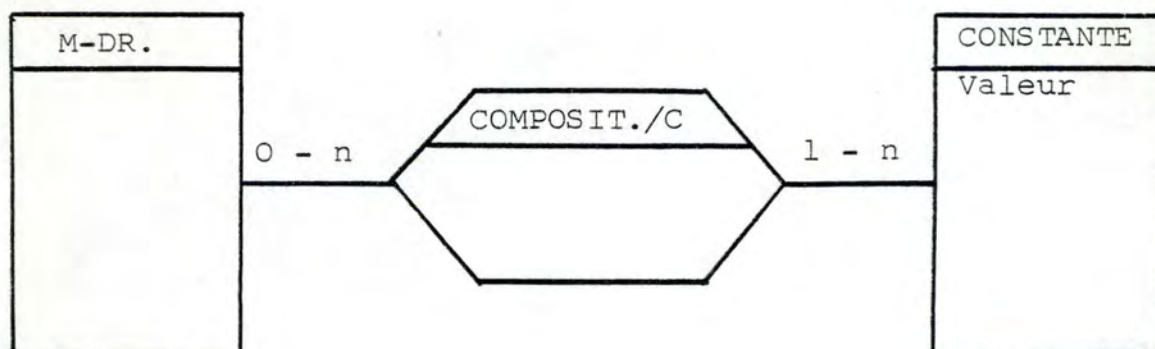
. DESCRIPTION DE LA RELATION "COMPOSITION/C" ENTRE UN MEMBRE DE DROITE ET UNE CONSTANTE.

- définition:

Une occurrence de cette relation signifie que le type d'information intervenant dans le membre de droite peut être exploité tel que: pas de recherche de valeurs, ni de génération d'équations.

Un membre de droite peut contenir 0 ou plusieurs constantes:
Une constante appartient toujours à un membre de droite; elle peut intervenir dans plusieurs membres de droite différents.

- représentation graphique:



I.2.2.6. DESCRIPTION DE L'ÉLÉMENT "TYPE-DONNÉE-DISPONIBLE".

a) définition: un type de donnée disponible est un type d'information dont les occurrences (valeurs) sont contenues dans la B.D. du S.I.A.D.

N.B.: cet élément n'est introduit que temporairement dans le schéma conceptuel du système d'équations. En effet, c'est un "élément tampon" entre le système d'informations et le système d'équations.

b) propriétés: dans le contexte du système d'équations, nous lui assignons 3 propriétés:

- un nom pouvant l'identifier.
- une définition.
- un format.

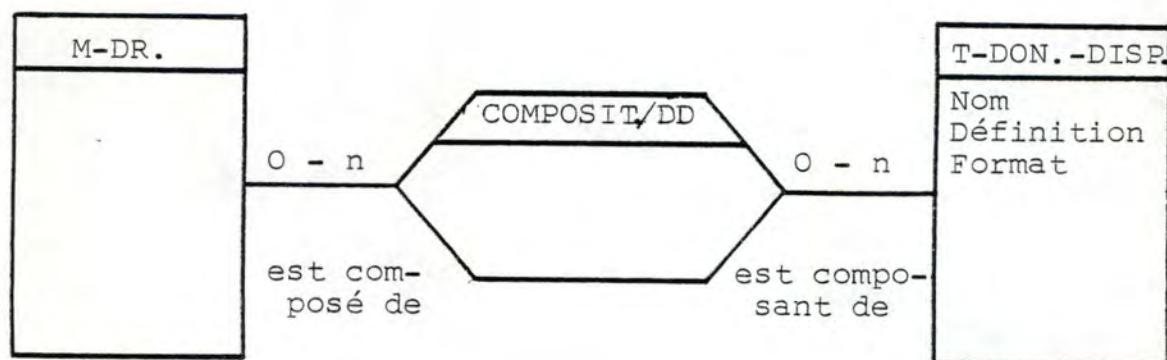
c) relation:

- . DESCRIPTION DE LA RELATION "COMPOSITION/DD" ENTRE UN TYPE DE DONNÉE DISPONIBLE ET UN MEMBRE DE DROITE.

- définition:

Une occurrence de cette relation signifie que le type d'information entrant dans la composition du membre de droite ne doit pas être calculé; les valeurs que prennent les occurrences de ce type sont disponibles dans la B.D.

Un membre de droite peut contenir 0 ou plusieurs types de donnée disponible. Un type de donnée disponible peut n'être exploité par aucune équation.

- représentation graphique:

I.2.2.7. DESCRIPTION DE L'ELEMENT "PARAMETRE".

a) définition: un paramètre est un type d'information ni calculé, ni enregistré dans la B.D. du S.I.A.D.
Ce sont des types "d'informations" extérieurs aux frontières du système d'informations de l'entreprise.

ex.: COURS DU DOLLARD
MONTANT D'UN SUBSIDE

Ils peuvent cependant être nécessaires dans le processus de calcul d'informations de gestion.

b) propriétés: les propriétés d'un paramètre sont:

- son nom.
- sa définition.
- son format.

c) relation:

- . DESCRIPTION DE LA RELATION "COMPOSITION/P" ENTRE UN MEMBRE DE DROITE ET UN PARAMETRE.

- définition:

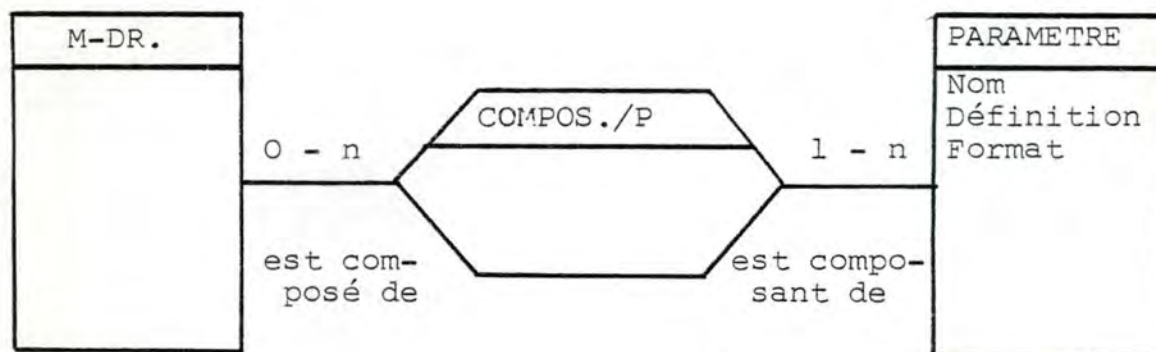
Une occurrence de cette relation signifie que le type d'information entrant dans la composition du membre de droite ne doit pas être calculé.

L'assignation d'une valeur à ce type d'information doit être demandée à l'utilisateur.

Un membre de droite peut contenir 0 ou plusieurs paramètres.

Un paramètre appartient toujours à un membre de droite; il peut intervenir dans plusieurs membres de droite.

- représentation graphique:



I.2.2.8. DESCRIPTION DE L'ELEMENT "APPEL-MODELE".

a) définition: puisque'un modèle peut fournir plusieurs "variables-résultats" différentes et qu'une même "variable-résultat" peut être fournie par des modèles différents, nous introduirons le concept "d'appel à un modèle pour un résultat" qui consiste à désigner univoquement une occurrence de la relation "RESULTAT (décrite précédemment).

Etant donné les limites du formalisme E/A, ce concept sera "assimilé" à un élément: "APPEL-MODELE", univoquement défini par une occurrence de "MODELE" et une occurrence de "VARIABLE" (participant à une occurrence de "RESULTAT" pour ce modèle).

ex.: soient les modèles:

"MNO" fournissant les variables-résultats "A(X)" et "B(X)"

"PQR" fournissant la variable-résultat "A(X)"

Les différents "appels-modèles" possibles seront:

- résultat "A(X)" du modèle "MNO"
- résultat "A(X)" du modèle "PQR"
- résultat "B(X)" du modèle "MNO"

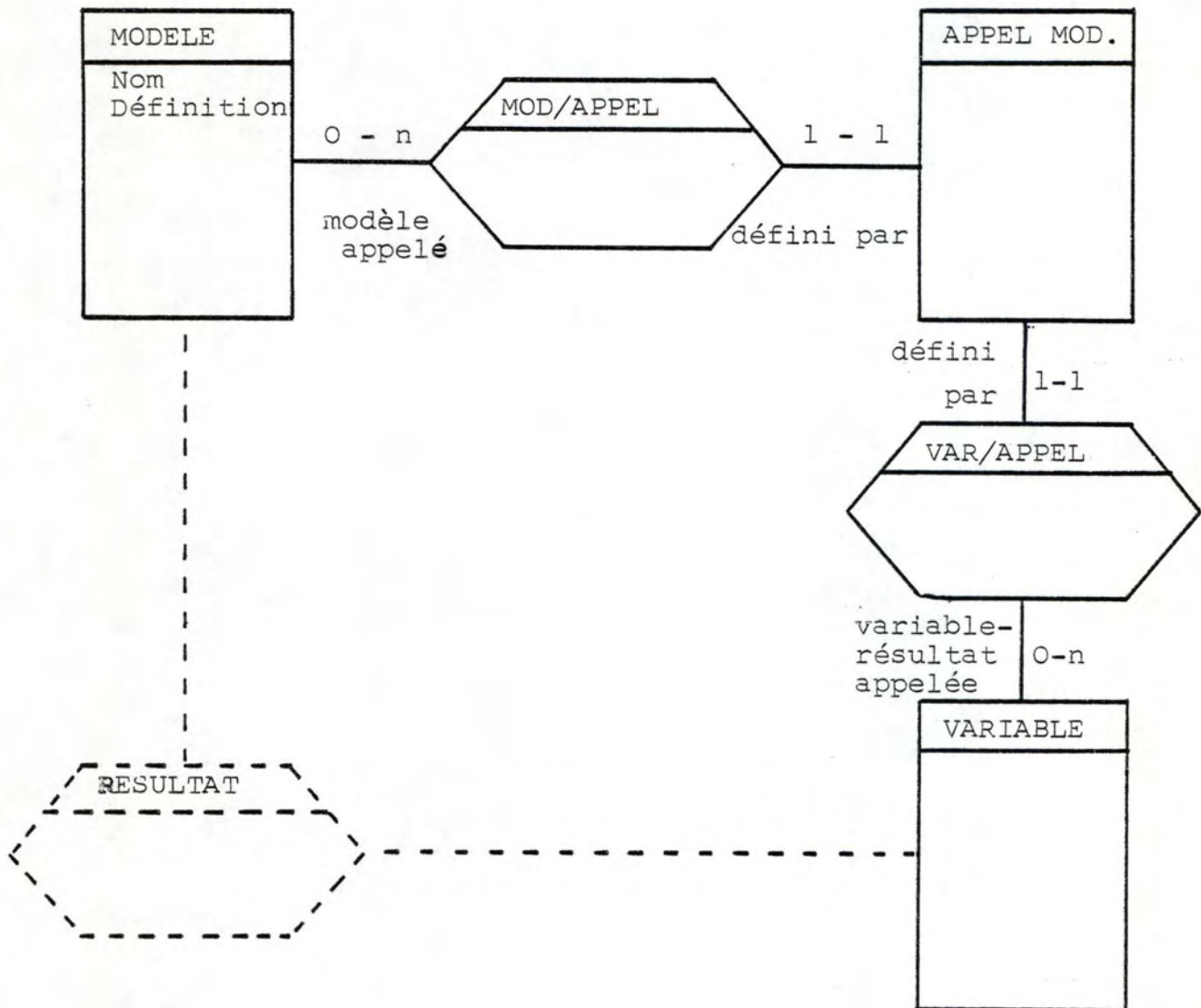
L'appel à un modèle est donc un "mécanisme de sélection" d'une variable-résultat et de son procédé de calcul (modèle).

b) propriété: l'élément "appel-modèle" n'a pas de propriétés intrinsèques*. Puisqu'il est assimilé à un élément, il aura un identifiant interne (géré par le système). Il est en fait identifié par une occurrence de MO-DELE et une occurrence de VARIABLE.

c) représentation graphique: Le concept "d'appel à un modèle pour un résultat" ne peut-être représenté graphiquement uniquement par l'élément "APPEL-MODELE".

En effet, cet élément est un élément "fictif" servant de tampons entre deux relations.

* Une propriété "intrinsèque" est une propriété "correspondant à un fait du monde réel"



Un APPEL-MODELE est défini par un et un seul modèle et une et une seule variable (participant à une occurrence de résultat pour ce modèle).

Un modèle peut intervenir dans 0 ou plusieurs "appels-modèle". Une variable peut intervenir dans 0 ou plusieurs "appels-modèle" (si elle y intervient, elle participe nécessairement à une occurrence de RESULTAT pour le modèle participant à l'occurrence MOD/APPEL de l'APPEL MODELE en question).

c) relation:

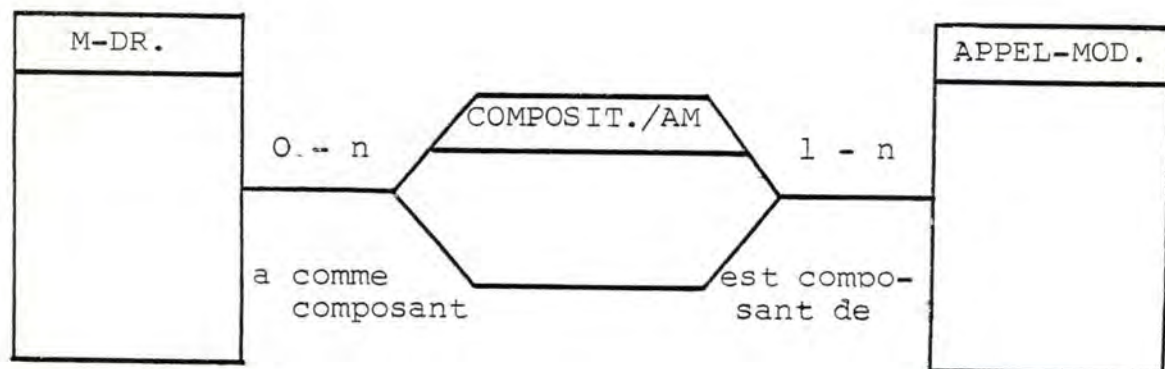
. DESCRIPTION DE LA RELATION "COMPOSITION/AM" ENTRE UN APPEL-MODELE ET UN MEMBRE DE DROITE.

- définition:

Une occurrence de cette relation signifie que le type d'information intervenant dans la composition des membres de droite, doit être calculé. Son processus de calcul est décrit par les équations générales (produisant le résultat en question) du modèle définissant l'appel.

Un membre de droite peut contenir 0 ou plusieurs appels-modèles. Un même appel peut intervenir dans plusieurs membres de droite.

- représentation graphique:



- contraintes:

Un modèle ne peut contenir d'équations dont le membre de droite contient un appel à lui-même (le modèle "appelant" doit être différent du modèle "appelé").

Un modèle ne peut contenir d'équations dont le membre de droite contient un appel à un autre modèle pour un résultat que le premier modèle fournit lui-même.

Aucune des équations fournissant le résultat "appelé" du modèle "appelant" ne peut avoir en membre de gauche, une variable également définie par une équation du modèle "appelant" (respect de la contrainte citée p.).

- remarque:

Le concept d'"appel à un modèle pour un résultat" a été introduit pour permettre à l'utilisateur d'intégrer entre eux des ensembles "cohérents" d'équations ("cohérent" signifiant: "décrivant, sans ambiguïté et de manière complète, le processus de calcul d'un type d'information").

exemple schématique:

Soient les modèles

- MNO constitué par des équations générales

$$M1: A(X) = B(X) + C(X)$$

$$M2: C(X) = B(X,Y) \times \dots$$

⋮

$$Mn: F(X,Y) = A(X) \times D(X,Y)$$

- PQR constitué des équations générales

$$P1: A(Y) = B(Y) \times C(Y)$$

⋮

$$Pn: H(X,Y) = G(Y) \times \underline{\text{RESULTAT}} F(X,Y) \underline{\text{DU MODELE}} \text{ MNO}$$

L'appel du modèle MNO par le modèle PQR (via l'équation P_n) signifie donc que les équations générales du modèle MNO, décrivant le procédé de calcul du résultat $F(X,Y)$, doivent être intégrées au modèle PQR, c'est-à-dire que le modèle PQR décrit ci-dessus, est équivalent au modèle PQR' formé des équations générales

$$\begin{aligned} M1: & A(X) = B(X) + C(X) \\ M2: & C(X) = B(X,Y) \times \dots \\ & \vdots \\ Mn: & F(X,Y) = A(X) \times D(X,Y) \\ P1: & A(Y) = B(Y) \times C(Y) \\ & \vdots \\ Pm: & H(X,Y) = G(Y) \times F(X,Y) \end{aligned}$$

(en supposant que toutes les équations du modèle MNO décrivent le processus de calcul du résultat $F(X,Y)$.)

Les deux modèles PQR et PQR' étant équivalents, le modèle PQR pourrait être représenté dans la base de connaissances par le modèle PQR'.

Le concept "d'appel à un modèle" pourrait donc ne pas être représenté dans le schéma conceptuel du système d'équations.

En effet, tout modèle contenant des équations formulées à l'aide de ce concept peut être représenté dans la base de connaissance par un modèle équivalent dans lequel ce concept n'apparaît pas. Le concept "d'appel à un modèle" pourrait être considéré uniquement comme un concept lié au langage de formulation d'équations.

Nous avons cependant choisi de le représenter car, pour l'utilisateur, la base de connaissances à une fonction documentaire: il est donc préférable que la structure d'intégration d'ensembles d'équations reste explicite.

I.2.2.9. DESCRIPTION DE L'ELEMENT "ARGUMENT".

Introduction.

La création d'un modèle peut être comparée à la création d'une procédure. Nous avons employé le terme "comparée" car le concept de modèle n'est pas identique au concept de procédure.

Un modèle n'est pas un ensemble d'instructions (une procédure, bien), mais en ensemble de DESCRIPTIONS d'équations particulières sous forme d'équations générales. Ces descriptions serviront à générer les équations particulières qui, elles, pourraient être appelées "instructions".

La création d'un modèle est pourtant comparable à la création d'une procédure: de la même manière qu'une procédure peut être appelée et exécutée une infinité de fois avec des valeurs d'arguments d'entrée différents, un modèle pourra être appelé une infinité de fois pour générer des équations. Les équations à générer seront déduites de la spécification de types d'information que nous appelons également "ARGUMENTS".

Illustration.

Supposons que certains des produits fabriqués soient vendus à l'étranger.

Le modèle qui suit décrit le procédé de calcul du bénéfice réalisé dans un certain pays pendant un mois donné. Il est construit sur base du système d'information suivant:



Le prix de vente d'un produit est fixé selon le pays dans lequel ce produit est vendu. Il est revu chaque mois. Le prix de revient d'un produit est connu par pays et par mois.

MODELE :

BENEFICE (PAYS/MOIS) =
 SOM(PRODUIT [VENTE] PAYS; BENEFICE(PRODUIT, PAYS/MOIS)).
 BENEFICE(PRODUIT, PAYS/MOIS) =
 QTTE-VENDUE(PRODUIT, PAYS/MOIS) x MARGE BENEFICIAIRE
 PRODUIT, PAYS/MOIS)
 MARGE-BENEFICIAIRE(PRODUIT, PAYS, MOIS) =
 PRIX-VENTE(PRODUIT, PAYS/MOIS) - PRIX-REVIENT(PRODUIT/MOIS).

Les arguments de ce modèle sont "PAYS" et "MOIS".

a) soit la spécification d'arguments: PAYS= "SUISSE"
 MOIS= "05/84"

De cette spécification d'arguments seront déduites les équations particulières:

- Bénéfice(Suisse/05-84) =
 bénéfice(P1, Suisse/05-84) + bénéfice(P3, Suisse/05-84)
 + bénéfice(P9, Suisse/05-84)
- Bénéfice(P1, Suisse/05-84) =
 QTTE-VENDUE(P1, Suisse/05-84) x MARGE BENEFICIAIRE
 (P1, Suisse/05-84).
- Bénéfice(P3, Suisse/05-84) =
- Bénéfice(P9, Suisse/05-84) =

→ 3 équations particulières déduites d'une même équation générale.

* le bénéfice réalisé dans un pays pendant un mois = somme sur tous les produits vendus dans ce pays, des bénéfices réalisés sur ces produits, dans ce pays, pendant un mois.

- Marge Bénéficiaire(P1,Suisse/05-84) =
prix-vente(P1,Suisse/05-84) - prix-revient(P1/05-84).
 - Marge Bénéficiaire(P3,Suisse/05-84) =
 - Marge Bénéficiaire(P9,Suisse/05-84) =
- 3 équations particulières déduites d'une même équation générale.

En supposant que la firme ne vend que les produits P1, P3, P9 en Suisse à cette époque.

La figure 10 est une représentation schématique de la similitude existant entre la création d'une procédure et la création d'un modèle.

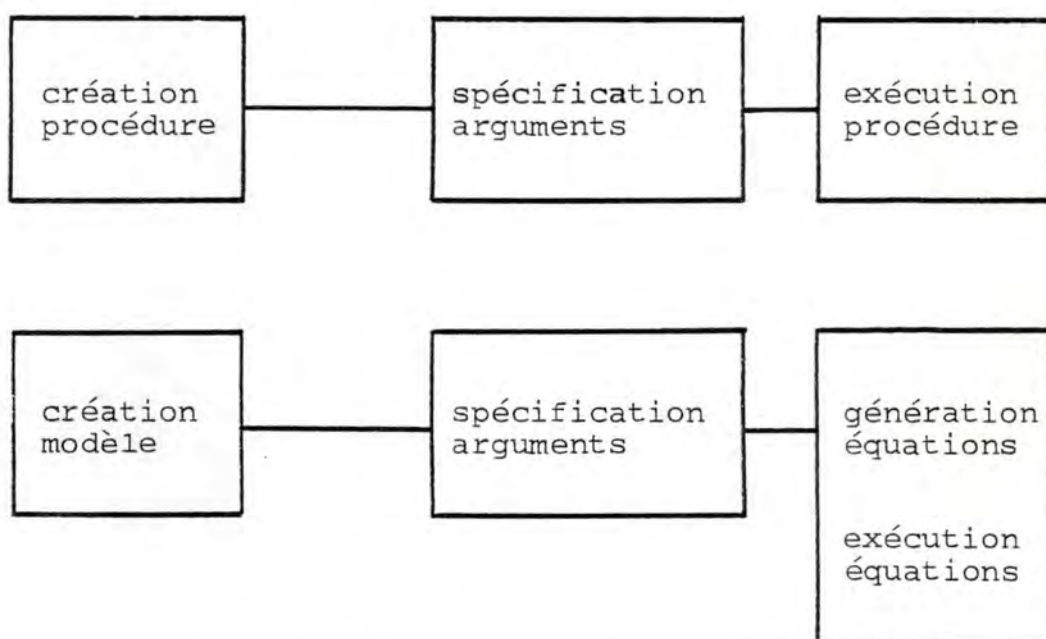


Figure 10 : similitude procédure-modèle.

a) définition: un argument est un type d'information dont la spécification permettra de générer les équations correspondant à un problème spécifique.

b) propriété: dans le contexte du système d'équations, nous attribuerons une seule propriété à l'élément "argument": un nom permettant de l'identifier.

c) relations:

. DESCRIPTION DE LA RELATION "ARG/MOD" ENTRE DES ARGUMENTS ET DES MODELES.

- définition:

Cette relation met en correspondance le modèle et ses arguments.

On admettra qu'un modèle puisse ne pas posséder d'arguments. Bien que les modèles décrits jusqu'à présent possédaient toujours au moins un argument "temporel" (JOUR,MOIS,AN), on peut imaginer qu'un tel système puisse fonctionner à partir de données non historiques (reflétant l'instant courant).

Dans ce cas, le modèle décrit par les équations:

$$\text{BENEFICE TOTAL} = \text{SOM}(\text{PAYS}; \text{BENEFICE}(\text{PAYS}))$$

$$\text{BENEFICE}(\text{PAYS}) = \text{SOM}(\text{PRODUIT}[\text{VENTE}]\text{PAYS}; \text{BENEFICE}(\text{PRODUIT}, \text{PAYS})).$$

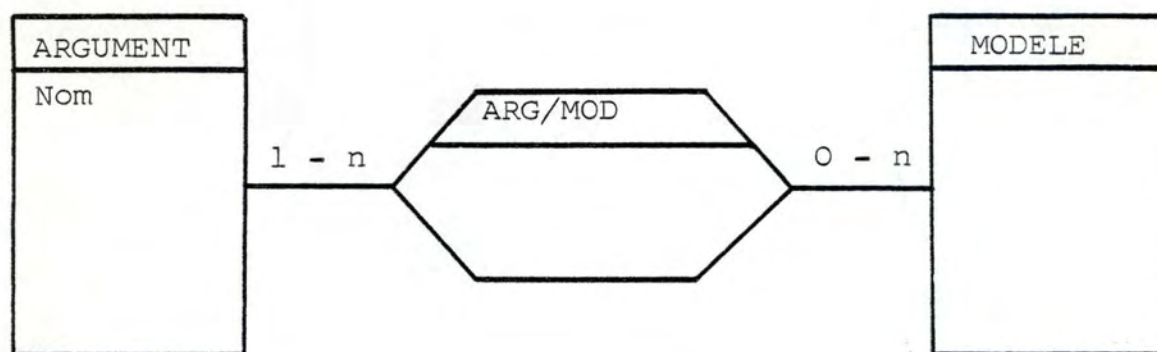
$$\text{BENEFICE}(\text{PRODUIT}, \text{PAYS}) = \text{QTTE-VENTUE}(\text{PRODUIT}, \text{PAYS}) \times \text{MARGE BENEFICIAIRE}(\text{PRODUIT}, \text{PAYS})$$

$$\text{MARGE BENEFICIAIRE}(\text{PRODUIT}, \text{PAYS}) = \text{PRIX-VENTE}(\text{PRODUIT}, \text{PAYS}) - \text{PRIX-REVIENT}(\text{PRODUIT}).$$

ne posséderait pas d'arguments.

Un même type d'information peut être argument de modèles différents.

- représentation graphique:



-oOo-

LE SCHEMA CONCEPTUEL INTEGRANT LES DIFFERENTS
CONCEPTS DECRITS EST REPRESENTÉ PAR LA FIGURE 11.

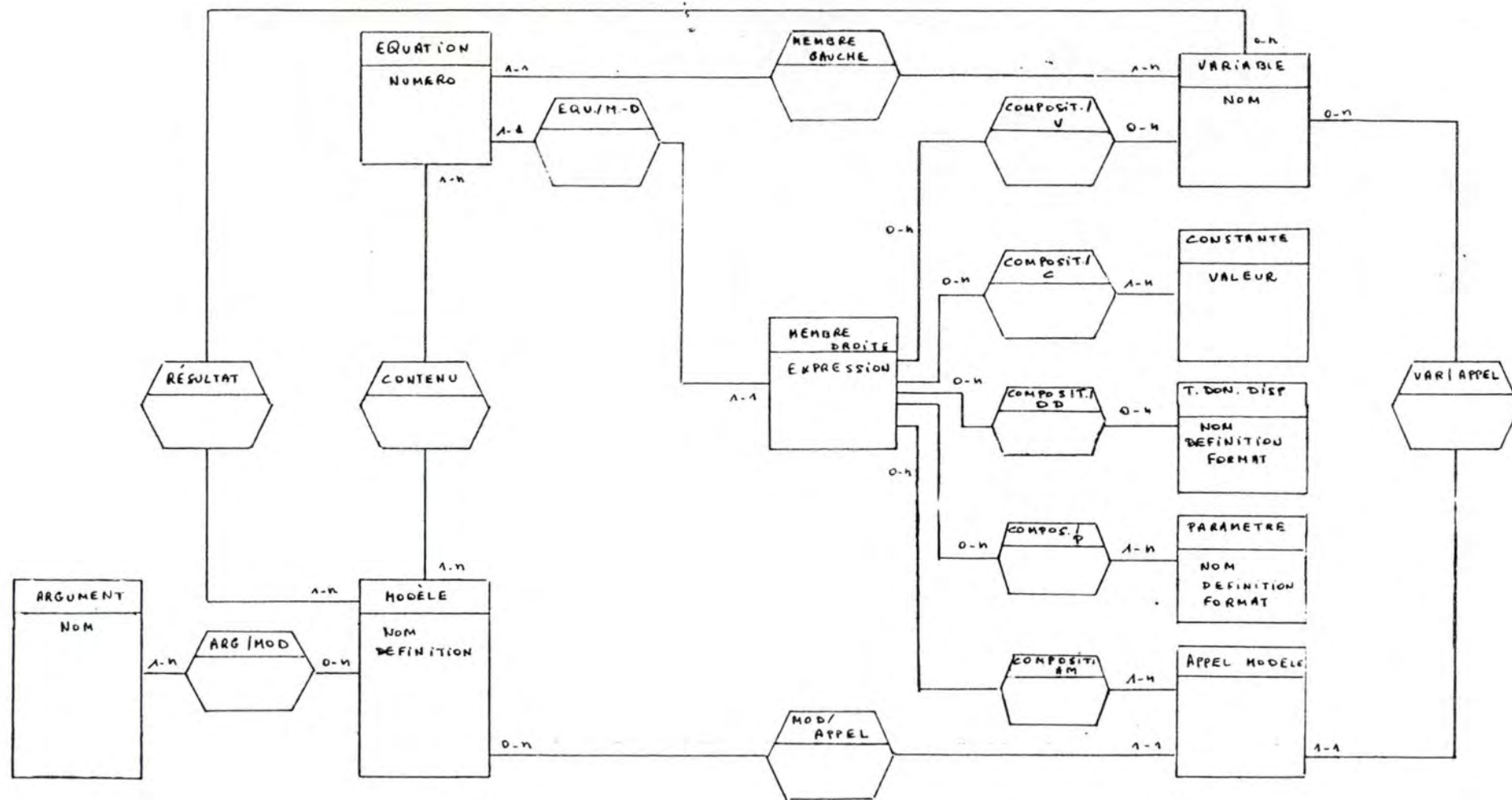


FIGURE 11.

I.2.3. ETUDES D'EXEMPLES SIMPLES ET CONCLUSIONS.

L'élaboration du schéma conceptuel du système d'équations décrit a été guidée par une question:

"QUELS SONT LES CONCEPTS A METTRE EN EVIDENCE POUR UNE DESCRIPTION, SOUS FORME D'EQUATIONS GENERALES, DU PROCEDE DE CALCUL DE TYPES D'INFORMATIONS ?".

L'étude d'exemples simples,
 le souci - de pouvoir prendre en charge des problèmes assez généraux,
 - de permettre à l'utilisateur d'intégrer entre eux des ensemble "cohérents" d'équations ("appel à un modèle").
 nous ont conduit à l'élaboration du schéma décrit jusqu'ici.

L'analyse s'est poursuivie par l'imagination de problèmes simples pouvant se poser à un gestionnaire, et par la description de leur processus de résolution à l'aide d'équations générales.

Ces problèmes ont été imaginés de manière à ce que les équations générales puissent reprendre tous les concepts mis en évidence dans le schéma.

Cette étude nous a conduit à conclure à la nécessité de la modification du schéma conceptuel du système d'équations. Cette conclusion peut être justifiée par l'analyse des modèles 1. et 2. décrits dans le chapitre suivant. Elle est dûe essentiellement à l'utilisation du concept d'appel à un modèle.

Analyse d'exemples.

L'exemple 2. a été imaginé de manière à ce que les équations générales décrivant son processus de résolution utilisent le concept d'appel à un modèle.

Le modèle décrit dans cet exemple, calcule le coût moyen d'un kg de fruit à partir des achats effectués chez les différents fournisseurs pendant un an. Ce coût est calculé en tenant compte des réductions octroyées lors de chaque achat.

Le modèle 1. avait été conçu pour calculer, à partir d'une quantité achetée, spécifiée par l'utilisateur ("QTTE-ACHETEE" déclarée comme paramètre), le montant total à payer lors d'un achat de fruit chez un fournisseur en tenant compte des réductions octroyées lors de l'achat.

Remarque: "QTTE-ACHETEE" est déclarée comme paramètre car les valeurs qui lui sont attribuées doivent être nécessairement spécifiées par l'utilisateur: elles sont tout-à-fait indépendantes des "valeurs réelles" reflétant l'activité de l'entreprise (ce modèle est essentiellement un outil de simulation).

D'après la nature des problèmes pris en charge par ces deux modèles, il semble a priori évident que le processus de calcul décrit dans le modèle 1. puisse être utilisé par le modèle 2. (donc qu'une équation du modèle 2. puisse contenir un appel au modèle 1.).

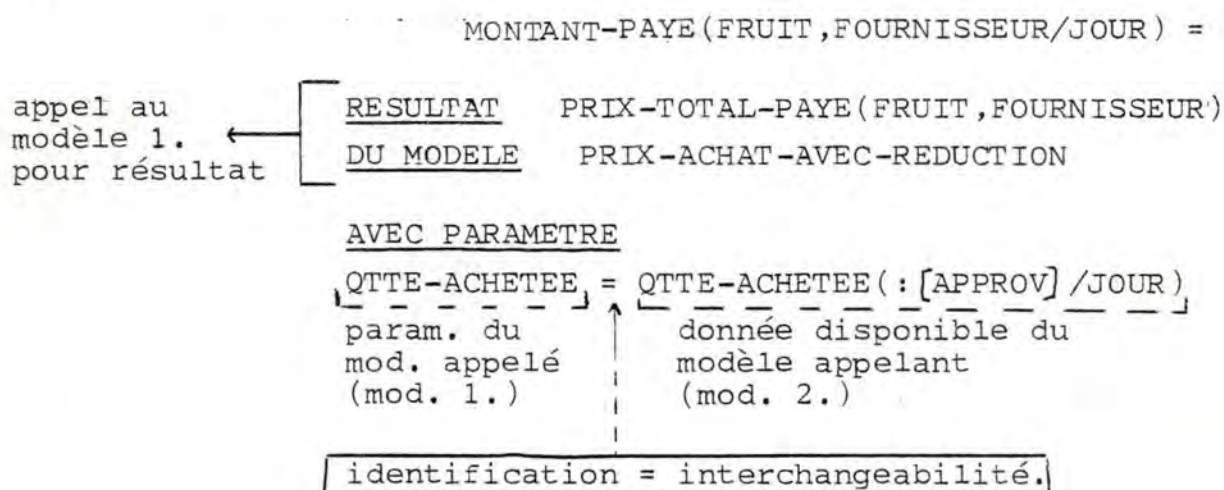
Un problème se pose cependant:

les équations générales composant le modèle 1. (et décrivant le processus de calcul qui pourrait être utilisé par le modèle 2.) ne sont pas compatibles avec le problème pris en charge par le modèle 2. En effet, elles font intervenir le paramètre "QTTE-ACHETEE".

Dans le contexte du modèle 2., ces équations devraient être modifiées: le type d'information "QTTE-ACHETEE" devrait être "remplacé" par un type d'information faisant référence aux données disponibles (le modèle 2. calcule le coût moyen d'un kg de fruits à partir d'achats réellement effectués: les valeurs des différentes quantités achetées sont disponibles dans la B.D. du S.I.A.D.).

Malgré ce problème, puisque le processus de calcul décrit par le modèle 1. est valable pour le modèle 2., nous admettons qu'il puisse contenir un appel à ce modèle 1.

Illustrons-en les conséquences par l'interprétation de l'équation du modèle 2. contenant cet appel.



Cet appel signifie que les équations générales du modèle 1. décrivant le processus de calcul du résultat "PRIX-TOTAL-PAYE (FRUIT, FOURNISSEUR/JOUR)" sont utilisables par le modèle 2. si le type de donnée disponible "QTTE-ACHETEE (: APPROV /JOUR)" y joue le rôle de paramètre "QTTE-ACHETEE". Si tel est le cas, la variable "MONTANT-TOTAL-PAYE (FRUIT, FOURNISSEUR/JOUR)" y joue le rôle de la variable-résultat PRIX-TOTAL-PAYE (FRUIT, FOURNISSEUR).

L'utilisation du concept d'appel à un modèle nous oblige donc à admettre que, dans le contexte du membre de droite d'une équation générale, des types d'information de nature "PARAMETRE" et "TYPE-DONNEE-DISPONIBLE" sont interchangeables.

Elles peuvent donc être représentées par un même élément.

C'est selon le modèle dans lequel sont intégrées les équations que sera distinguée la nature "paramètre" ou "donnée disponible" du type d'information représenté par cet élément.

REMARQUE.

C'est volontairement que nous avons présenté le schéma conceptuel "intermédiaire". En effet, ce n'est que grâce à la formulation de la méthode de résolution de problèmes "concrets", sous forme d'équations générales reprenant les différents concepts décrits dans le premier schéma, qu'est apparue la nécessité d'une modification.

Le schéma conceptuel résultant de cette modification n'était donc justifiable que par rapport au schéma conceptuel "intermédiaire".

I.2.4. MODIFICATION DU SCHEMA CONCEPTUEL DU SYSTEME D'EQUATIONS.

Ce n'est donc plus dans le contexte du membre de droite d'une équation que nous devons distinguer les natures "PARAMETRE" ou "TYPE-DONNEE-DISPONIBLE" d'un type d'informations non calculé mais dans le contexte d'un modèle ;

Au niveau du membre de droite d'une équation, nous distinguerons les types d'information de nature:

"VARIABLE"

"CONSTANTE"

"APPEL A UN MODELE"

"DONNEE"

Une "DONNEE" sera un élément intermédiaire, représentant des types d'information interchangeable. Une donnée représentera toujours un type d'information non calculé (dont la valeur sera, selon le modèle, à rechercher dans la B.D. ou à demander à l'utilisateur).

Nous créerons une relation entre une donnée et un modèle. Cette relation possèdera une propriété: le statut de la donnée pour le modèle.

Une donnée fera référence à un "PARAMETRE" ou un "TYPE-DONNEE-DISPONIBLE", selon le statut qu'aura la donnée dans le modèle.

La description de ces nouveaux éléments et relations suit le schéma rectifié, représenté dans la figure 12.

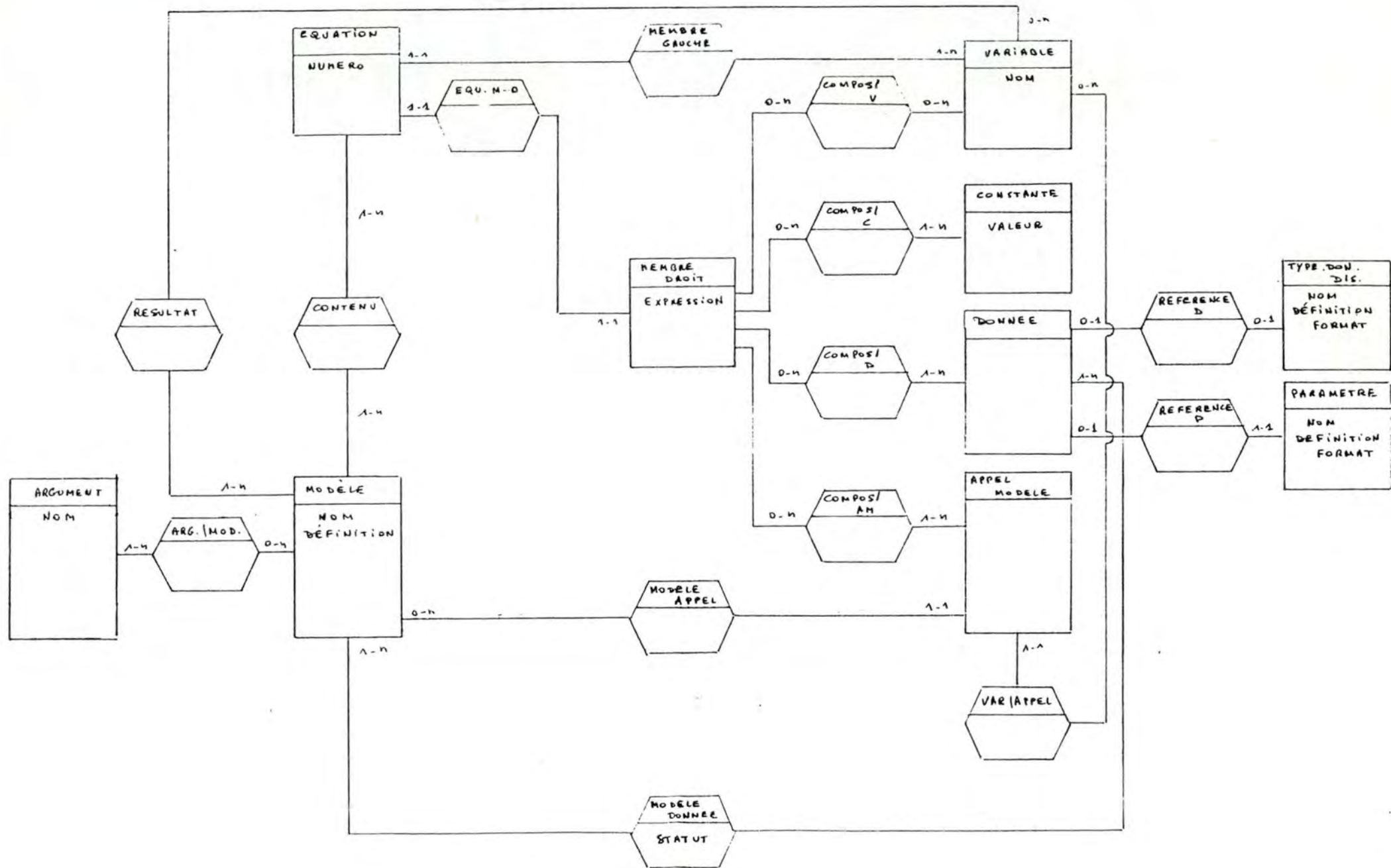


FIGURE 12

I.2.4.1. DESCRIPTION DE L'ELEMENT "DONNEE".

a) définition: une donnée représente des types d'information non calculés intervenant dans un modèle, soit sous la forme paramètre, soit sous la forme donnée disponible.

b) propriétés: une donnée n'a qu'une propriété : un identifiant interne (géré par le système).
(Le concept de "donnée" est inconnu pour l'utilisateur. C'est un élément intermédiaire géré par le système).

c) relations:

. DESCRIPTION DE LA RELATION "MODELE-DONNEE".

- définition:

Une occurrence de cette relation met en correspondance le modèle et les désignations des types d'information non calculés qui entrent dans la composition des équations qu'il contient.

Elle a une propriété: le statut de la donnée pour le modèle.

Un modèle est relié à au moins une donnée. En effet, les types d'information dont le modèle décrit le procédé de calcul doivent nécessairement "provenir" de types d'information non calculés.

Une donnée doit toujours être utilisée par un modèle.

. DESCRIPTION DE LA RELATION "REFERENCE-D".

- définition:

Cette relation met en correspondance la donnée et le type de donnée disponible à laquelle elle fait référence. Une donnée fait référence à 0 ou 1 type de donnée disponible (à 0, car elle peut faire uniquement référence à un paramètre). Un type de donnée disponible peut être référencé par 0 ou une donnée (0, car certains types de donnée disponible peuvent ne pas intervenir dans le calcul d'autres types d'information).

- contraintes:

Une occurrence de cette relation n'existe que si la propriété "statut" de la relation "MODELE-DONNEE" = "D" (avec "D" étant la valeur du code représentant le statut "donnée-disponible").

. DESCRIPTION DE LA RELATION "REFERENCE-P".

- définition:

Cette relation met en correspondance la donnée et le paramètre auquel elle fait référence. Une donnée fait référence à 0 ou 1 paramètre(s) (0, car elle peut faire uniquement référence à un type-donnée-disponible. Un paramètre est représenté par une et une seule donnée.

-contrainte:

Une occurrence de cette relation n'existe que si la propriété "statut" de la relation "MODELE-DONNEE" = P.

Une donnée participe toujours à une des deux relations "REFERENCE-D" ou "REFERENCE-P".

Ces deux relations ne sont pas mutuellement exclusives car une même "donnée" peut représenter à la fois un "type-donnée-disponible" et un paramètre (cas de l'illustration modèle 1. et 2.).

--o00o--

I.3. INTEGRATION DES SOUS-SCHEMAS CONCEPTUELS DU SYSTEME D'INFORMATION ET DU SYSTEME D'EQUATIONS.

I.3.1. IDENTITE DES ELEMENTS "TYPE-DONNEE-DIPONIBLE" ET "ATTRIBUT".

Les "types-données-diponibles" apparaissant dans le membre de droite d'une équation (via l'élément "donnée" sont en fait des attributs).

En effet, l'exécution des équations générées ne peut se faire que par l'attribution de valeurs aux informations (occurences de données disponibles) qu'elles contiennent.

La relation "REFERENCE-D" devient donc une relation de "REQUETE" à un "ATTRIBUT".

Contrainte_d'intégrité:

Seuls les attributs à valeur numérique participent aux occurences de l'association REQUETE.

I.3.2. IDENTITE DES ELEMENTS "ARGUMENTS" ET "TYPE-ENTITE" OU "TYPE-PERIODE".

Les arguments d'un modèle sont toujours des types d'entité ou des types de période. La spécification des arguments d'un modèle consiste donc à sélectionner les occurences de ces types de manière à pouvoir générer les équations correspondantes.

La relation "ARG-MOD" sera donc scindée en deux relations (puisque nous avons distingué les "types d'entités" des "types de période").

I.3.3. DESCRIPTION DE LA RELATION "CORRESPONDANCE" ENTRE UNE "VARIABLE" ET UN "ATTRIBUT".

Certains types d'information, membres de gauche d'équations, sont des caractéristiques de T.E. ou T.A.

L'exécution de modèles généraux peut donc servir à compléter la B.D. du S.I.A.D.

- définition:

Une occurrence de la relation "CORRESPONDANCE" signifie que la variable correspond à un attribut dont les valeurs, calculées grâce aux exécutions* de modèles, doivent être enregistrées dans la B.D. du S.I.A.D.

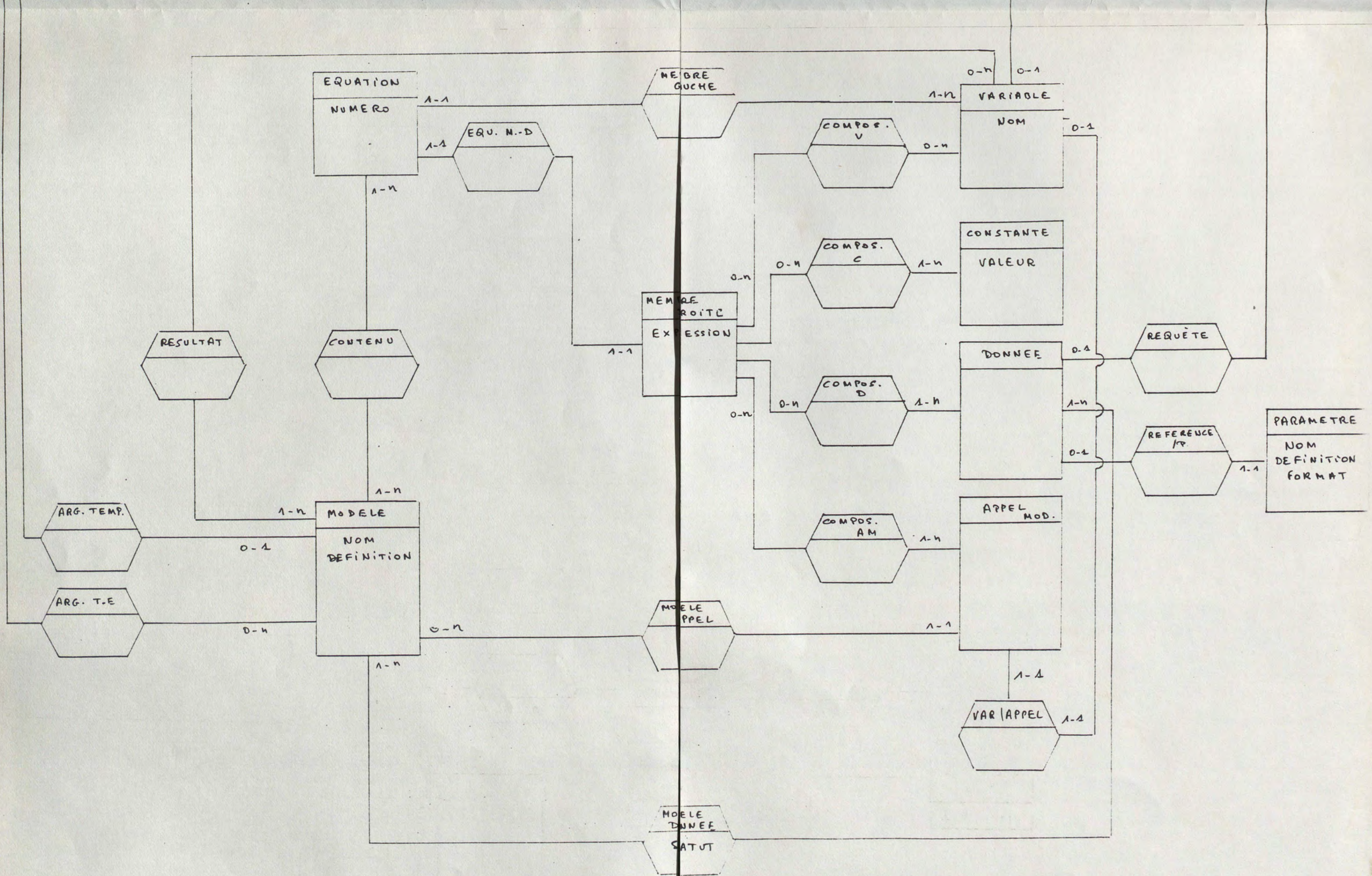
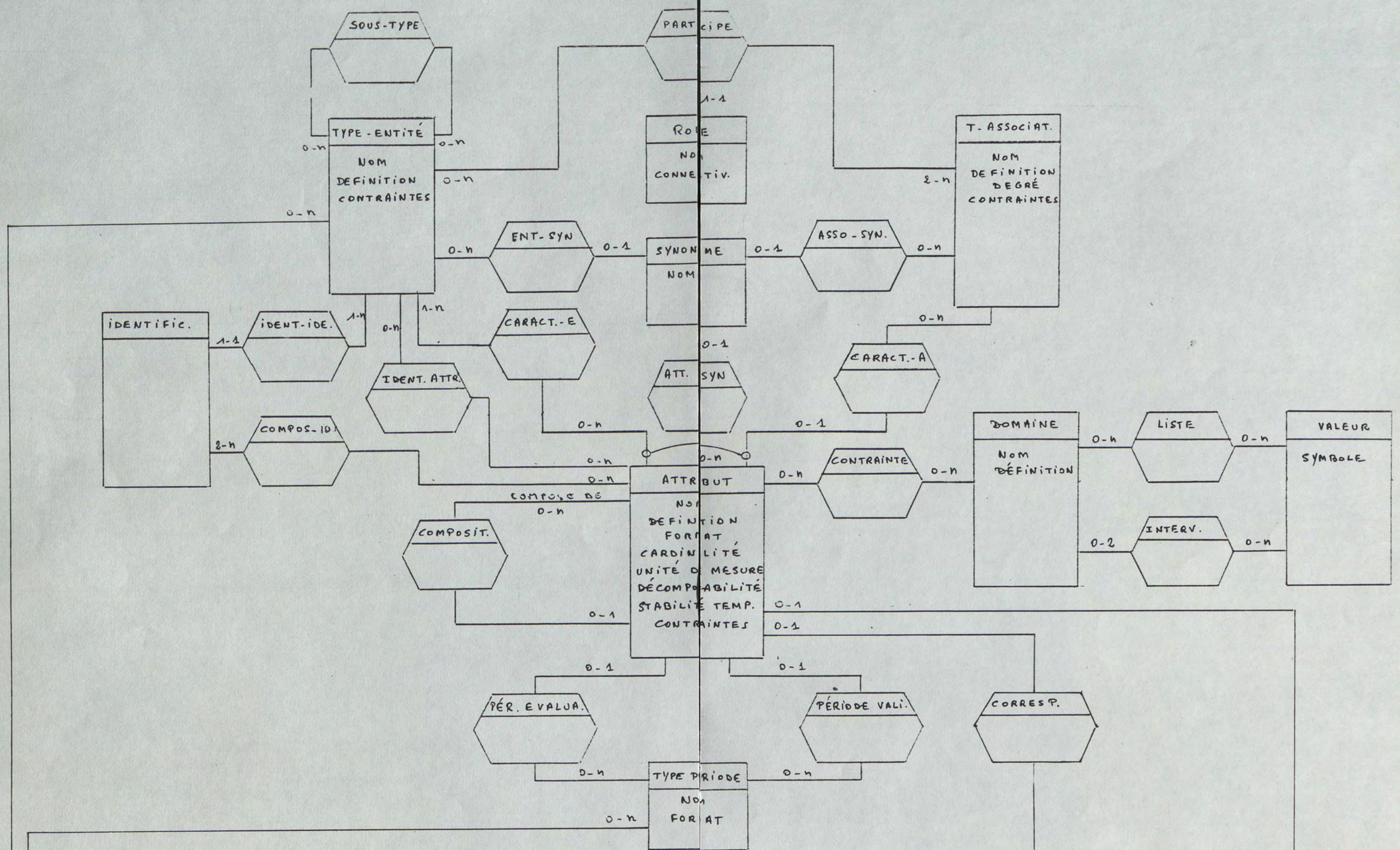
A un attribut correspond 0 ou 1 variable; 0, car les "attributs non calculés" ne correspondent à aucune variable.

Une variable s'identifie à au plus un attribut.

--oOo--

LE SCHEMA INTEGRE EST REPRESENTÉ DANS LA FIGURE 13.

* Nous appelons "exécution d'un modèle général" les phases de génération d'équations et d'exécution de ces équations.



CHAPITRE II

UN LANGAGE DE SPECIFICATION DE MODELES GENERAUX.

Ce chapitre est consacré à la présentation d'un langage de spécifications de modèles généraux.

Nous décrirons tout d'abord les caractéristiques générales de ce langage. Nous l'illustrerons ensuite au moyen d'exemples de problèmes qui devraient pouvoir être pris en charge par le générateur de modèle.

II.1. CARACTERISTIQUES DU LANGAGE.

II.1.1. Ce langage est NON-PROCEDURAL.

Devant être utilisé par des non-informaticiens, il est indispensable que ce langage soit non-procédural.

La spécification d'un modèle sera composée d'une partie déclarative et d'une partie "formulation d'équations".

Dans la partie déclarative seront spécifiés:
 le nom du modèle, sa définition, ses résultats, ses arguments
 et les paramètres qu'il utilise.
 Outre le fait que cette partie doit commencer par la déclaration
 du nom du modèle, les autres déclarations peuvent être faites
 dans n'importe quel ordre.

La partie "formulation d'équations" sera séparée de
 la partie déclarative par un mot réservé (DEBUT). Les équations
 générales peuvent également être spécifiées dans n'importe quel
 ordre.

II.1.2. CE LANGAGE EST BASE SUR LE PRINCIPE DE QUALIFICATION.

Le principe de qualification permet d'obtenir un langage
 relativement naturel.

Nous imposerons que le nom d'un T.E. définisse univoque-
 ment ce T.E. et que le nom d'un T.A. définisse univoquement
 ce T.A..

Ces noms pourront donc servir de qualificatifs

- pour les attributs qui caractérisent ces T.E. ou ces
 T.A.
- pour les types d'information calculés qui caractéri-
 sent ces T.E. ou ces T.A..

Un nom d'attribut ou de variable pourra donc ne pas
 être univoque. L'ensemble complet du (des) qualificatifs) de ce
 nom permettra de désigner univoquement un attribut ou une vari-
 able.

Exemple:

Soit le T.E. "PRODUIT".

Pour désigner le prix du produit nous qualifions le nom "prix" par "produit": PRIX(: PRODUIT).

Nous pouvons également avoir PRIX(:MAT-PREMIERE)

Les attributs et types d'information calculés seront également qualifiés, si nécessaire, par un type de période (de validité ou d'évaluation selon le cas). Cette qualification temporelle est mise en évidence par l'utilisation du "/".

Exemple:

QTTE-VENTUE(:PRODUIT/MOIS)

PRIX(:PRODUIT/MOIS) (en vigueur pendant le mois).

Cette contrainte est justifiée en annexe.

Les attributs ou caractéristiques calculés de T.A. seront qualifiés soit par le T.A., soit par les T.E. que ce T.A. met en correspondance et également (si nécessaire) par un type de période.

Exemple:

QTTE-ACHETEE(: [APPROVISIONNEMENT]/MOIS)

↔ QTTE-ACHETEE(:FRUIT, FOURNISSEUR /MOIS)

si APPROVISIONNEMENT est un T.A. mettant en correspondance les T.E. FRUIT et FOURNISSEUR

Remarque:

L'utilisation du ":" distingue les types d'information disponibles dans la B.D. du S.I.A.D. des types d'information calculés.

II.1.3. CE LANGAGE EST UN LANGAGE DE TYPE "NAVIGATIONNEL".

Ceci signifie qu'il est basé sur le parcours de graphes. Le "graphe" utilisé ici est l'occurrence du méta-schéma relatif au système d'information.

Exemple:

- BENEFICE (DEPARTEMENT/MOIS)
 = SOM(PRODUIT[VENTE]DEPARTEMENT, BENEFICE PRODUIT /MOIS)
 (= la somme sur tous les produits liés au département par VENTE).
- Un autre exemple de cette caractéristique peut être trouvé dans l'exemple 3 (emploi du mot réservé "POUR").

La formulation d'équations générales doit être supportée par la représentation graphique du système d'information.

II .2. EXEMPLE 1.

II .2.1. EXPLICATIONS.

Une firme de fabrication de confitures et conserves utilise des fruits comme matières premières. Elle s'approvisionne chez différents fournisseurs. Le prix d'achat (F/kg) des fruits est propre à chaque fournisseur. Il reste constant durant toute la saison (chaque année le fournisseur revoit ses prix). A chaque achat, les différents fournisseurs octroient des réductions, selon la quantité achetée.

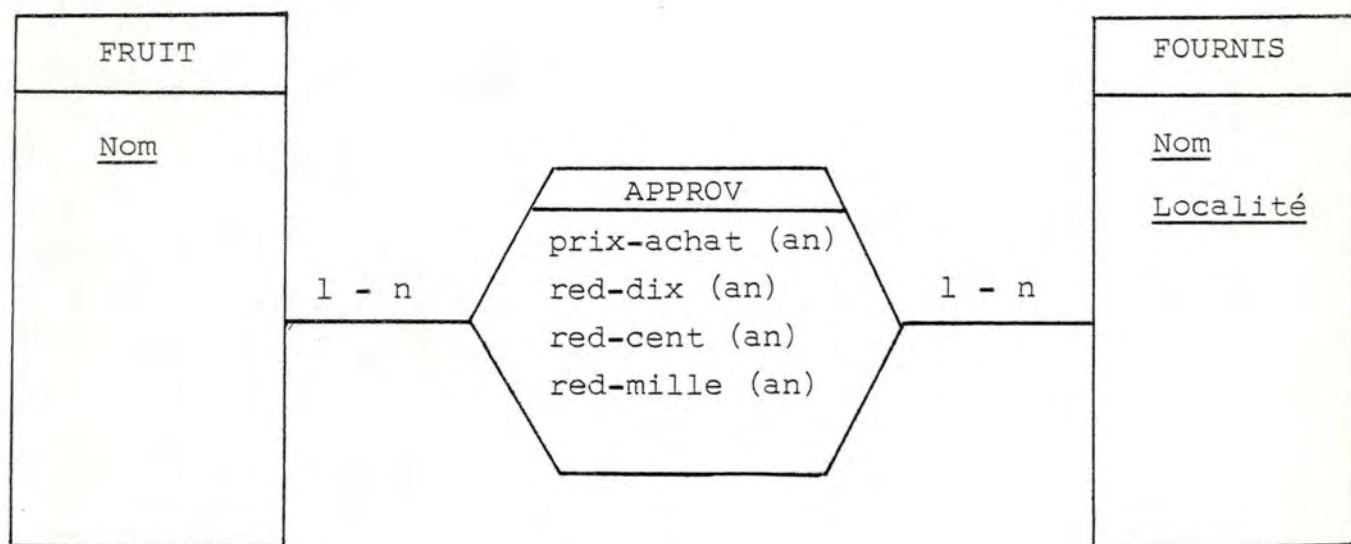
Il existe 3 tarifs de réduction: tarif de réduction par 10 kg, tarif de réduction par 100 kg, tarif de réduction par 1000 kg.

Ces différents tarifs de réduction dépendent également du fruit acheté.

Le gestionnaire, désireux de pouvoir sélectionner le fournisseur le plus favorable lorsqu'il doit passer une commande, veut évaluer rapidement le prix d'achat réel (avec réduction) du kg de fruits, selon le fournisseur choisi et la quantité achetée.

II .2.2. SYSTEME D'INFORMATION.

Le système d'information correspondant à ce problème peut être représenté graphiquement comme suit:



Il peut être décrit par deux T.E. "Fruit" et "Fournisseur" liés par le T.A. "Approvisionnement":

- un fruit est identifié par son NOM.
- un fournisseur est identifié par son NOM et sa LOCALITE.
- aucun autre attribut ne caractérise ces T.E.
- Les attributs du T.A. "Approvisionnement" sont: "PRIX-ACHAT", "RED-DIX", "RED-CENT", "RED-MILLE (les 3 taux de réduction).
- à ces attributs est associée une même période de validité de type "ANNEE".

II .2.3. SYSTEME D'EQUATIONS.

Le modèle correspondant à ce problème sera déclaré de la manière suivante:

<u>MODELE</u>	PRIX-ACHAT-AVEC-REDUCTION.
<u>DEFINITION</u>	Pour un fruit et un fournisseur, le modèle permet de calculer le prix d'achat réel d'un kg de fruits à partir d'une quantité achetée à spécifier.
<u>RESULTAT</u>	PRIX-TOTAL-PAYE(FRUIT,FOURNISSEUR), PRIX-ACHAT-REEL-KG(FRUIT,FOURNISS.).
<u>ARGUMENTS</u>	FRUIT, FOURNISSEUR , ANNEE.
<u>PARAMETRE</u>	QTTE-ACHETEE, <u>FORMAT</u> N; 4.
<u>DEBUT</u>	
⋮	voir équations page suivante.
<u>FIN.</u>	

II .2.4. STRUCTURES DES TABLEAUX.

Les figures 13 et 14 représentent deux types de tableaux pouvant être générés à partir de ce modèle.

Ces tableaux présentent toutes les informations intervenant dans le calcul.

La figure 15 représente un type de tableau dans lequel toutes les informations ne sont pas présentes: (décomposition de "quantité achetée" en tonnes, quintaux, ...). En effet, ces informations ne sont pas très utiles: elles surchargent le tableau.

La nature des informations dérivées des informations non représentées est décrite par la mention "CALCULE" (de manière non visible dans le tableau).

DEBUT

$\text{PRIX-ACHAT-REEL-KG}(\text{FRUIT}, \text{FOURNISSEUR}) = \text{PRIX-TOTAL-PAYE}(\text{FRUIT}, \text{FOURNISSEUR}) / \text{QTTE-ACHETEE}.$

$\text{PRIX-TOTAL-PAYE}(\text{FRUIT}, \text{FOURNISSEUR}) = \text{PRIX-ACHAT}(: [\text{APPROV.}] / \text{ANNEE}) \times \text{QTTE-ACHETEE} - \text{REDUCTION}.$

$\text{REDUCTION} = \text{REDUCTION-TONNE} + \text{REDUCTION-QUINTAL} + \text{REDUCTION-DIZAIN}.$

$\text{REDUCTION-TONNE} = \text{NBRE-TONNE} \times \text{RED-MILLE}(: [\text{APPROV.}] / \text{ANNEE}).$

$\text{REDUCTION-QUINTAL} = \text{NBRE-QUINTAL} \times \text{RED-CENT}(: [\text{APPROV.}] / \text{ANNEE}).$

$\text{REDUCTION-DIZAIN} = \text{NBRE-DIZAIN} \times \text{RED-DIX}(: [\text{APPROV.}] / \text{ANNEE}).$

$\text{NBRE-TONNE} = \text{ARRONDI-INF}(\text{QTTE-ACHETEE} / 1000).$

$\text{NBRE-QUINTAL} = \text{ARRONDI-INF}((\text{QTTE-ACHETEE} - (\text{NBRE-TONNE} \times 1000)) / 100).$

$\text{NBRE-DIZAIN} = \text{ARRONDI-INF}((\text{QTTE-ACHETEE} - (\text{NBRE-TONNE} \times 1000 + \text{NBRE-QUINTAL} \times 100)) / 10).$

FIN.

	A	B	C	D	E
1.	<u>FRUIT</u>	ARGUMENT			
2.	<u>FOURNISS.</u>	ARGUM.	ARGUM.		
3.	<u>PRIX-ACHAT</u>	<u>RED-DIX</u>	<u>RED-CENT</u>	<u>RED-MILLE</u>	
4.	INTR.AUTOM.	INTR.AUTOM.	INTR.AUTOM.	INTR.AUTOM.	
5.	<u>QTTE-ACHET.</u>	<u>NBRE-DIZ.</u>	<u>NBRE-QUINT.</u>	<u>NBRE-TONNE</u>	
6.	VAL. SPECIF.	CALCULE	CALCULE	CALCULE	
7.		<u>REDUC.DIZ.</u>	<u>REDUC.QUINT</u>	<u>REDUC.TONNE</u>	<u>REDUCTION</u>
8.		B4 x B6	C4 x C6	D4 x D6	B8 + C8 + D8
9.	<u>PRIX-TOT-PAYE</u>	<u>PRIX-ACHAT</u> <u>REEL-KG</u>			
10.	A4 x A6 - E8	A10 / A6			

Figure 13.

LEGENDE:

VAL. SPECIF.= valeur à spécifier

Intr. AUTOM.= introduction automatique

Les "labels" sont soulignés

Les flèches indiquent les données correspondant aux labels

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1.	FRUIT	FOURNISS.	QTTE-ACH.	PRIX-ACH.	RED-DIX	RED-CENT	RED-MILLE	NBRE-DIZ	NBRE-QUI	NBRE-TONNE	REDUC-DIZ	RED-QUIN	RED-TON	REDUCT	PRIX-TOT- PAYE	PRIX-ACHAT REEL FG
2.	ARGUM.	ARGUM.	VAL.SPECI	INTR-AUT.	INTR.AUT	INTR.AUT	INTR.AUT	CALC	CALC	CALC	E2 x H2	F2 x I2	G2 x J2	K2+L2+ M2	C2xD2- N2	O2 / C2
3.		ARGUM.														

Figure 14.

	A	B	C	D	E	F	G	H	I	J	K	L	M	
1.	FRUIT	FOURNISS	QTTE-ACH.	PRIX-ACH	RED-DIX	RED-CENT	RED-MILLE	REDUC-DIZ	REDUC-QUIN	REDUC-T	REDUCTION	PRIX-TOT PAYE	PRIX-ACH REEL/KG	
2.	ARGUM	ARGUM	VAL.SPECI	VAL.SPEC	VAL.SPEC	VAL.SPEC	VAL.SPEC	CALC	CALC	CALC	H2+I2-K2	C2xD2-K2	L2 / C2	
3.		ARGUM												

Figure 15.

II .2.5. EXECUTION DU MODELE.

Supposons que le gestionnaire dispose d'un langage de commande dont une primitive lui permet d'exécuter les modèles.

Exemple:

GEST: EXECUTER MODELE PRIX-ACHAT-AVEC-REDUCTION.

SYST: POUR QUEL FRUIT ? (NOM)

GEST: CERISE.

SYST: POUR QUEL FOURNISSEUR ? (NOM, LOCALITE)

GEST; DUPONT, LIEGE.

SYST: POUR QUELLE ANNEE ?

GEST: COURANTE.

SYST: VALEUR DE QTTE-ACHETEE ?

GEST: 550.

La description du modèle contenue dans la B.D. de spécifications permet au système de conduire le dialogue.

<u>ex.:</u> POUR QUEL FOURNISSEUR ?	(NOM, LOCALITE)
ARGUMENT	IDENTIFIANT DE L'ARGUMENT

Cette exécution est représentée à la figure 16.
La structure du tableau est celle de la figure 13.

	A	B	C	D	E
1.	<u>FRUIT</u>	CERISE			
2.	<u>FOURNISSEUR</u>	DUPONT	LIEGE		
3.	<u>PRIX-ACHAT</u>	<u>RED-DIX</u>	<u>RED-CENT</u>	<u>RED-MILLE</u>	
4.	25	25	300	3200	
5.	<u>QTTE-ACHETEE</u>	<u>NBRE-DIZ</u>	<u>NBRE-QUINT</u>	<u>NBRE-TONNE</u>	
6.	550	5	5	0	
7.		<u>REDUC-DIZ</u>	<u>REDUC-QUINT</u>	<u>REDUC-TONNE</u>	<u>REDUCTION</u>
8.		125	1500	0	1625
9.	<u>PRIX-TOT-PAYE</u>	<u>PRIX-ACHAT</u> <u>REEL-KG</u>			
10.	12.125	22,04			

Figure 16.

II .3. EXEMPLE 2.

II .3.1. EXPLICATIONS.

Le gestionnaire envisage de créer un modèle lui permettant de calculer rapidement le prix de revient de ses produits finis.

Ce calcul nécessitant la connaissance du coût du kilo de fruits, il crée donc tout d'abord le modèle qui lui permet d'évaluer le coût moyen du kilo de fruits (indépendamment du fournisseur).

Il désire que cette estimation soit réalisée à partir des différents achats effectués pendant l'année.

Le coût du kilo de fruits estimé sera donc "valide" pour une période de type "année".

II .3.2. SYSTEME D'INFORMATION.

L'exécution du modèle nécessite la connaissance des différentes quantités achetées pendant l'année.

Le gestionnaire étend donc le système d'information existant en ajoutant un attribut de nom "QTTE-ACHETEE" au T.A. "APPROVISIONNEMENT" défini entre "FRUIT" et "FOURNISSEUR".

Il assigne à cet attribut une période d'évaluation de type "JOUR", signalant que les différentes valeurs de quantités achetées de fruits chez un fournisseur représentent les quantités achetées à une certaine date.

La spécification de ce nouvel attribut pourrrait se faire, soit directement grâce au langage de spécification, soit de manière guidée (grille formatée à compléter ou présentations successives des différentes rubriques à compléter, ...)

E xemple de spécification guidée.

UTIL: DECRIRE ATTRIBUT

SYST: Nom attribut ?

UTIL: QTTE-ACHETEE

SYST: Attribut de ? T.E. nom T.E. ou T.A. nom T.A.

UTIL: T.A. APPROVIS.

SYST: Stabilité temporelle?

Stable ? (S)

Instantané ? (I)

A période d'évaluation ? (P)

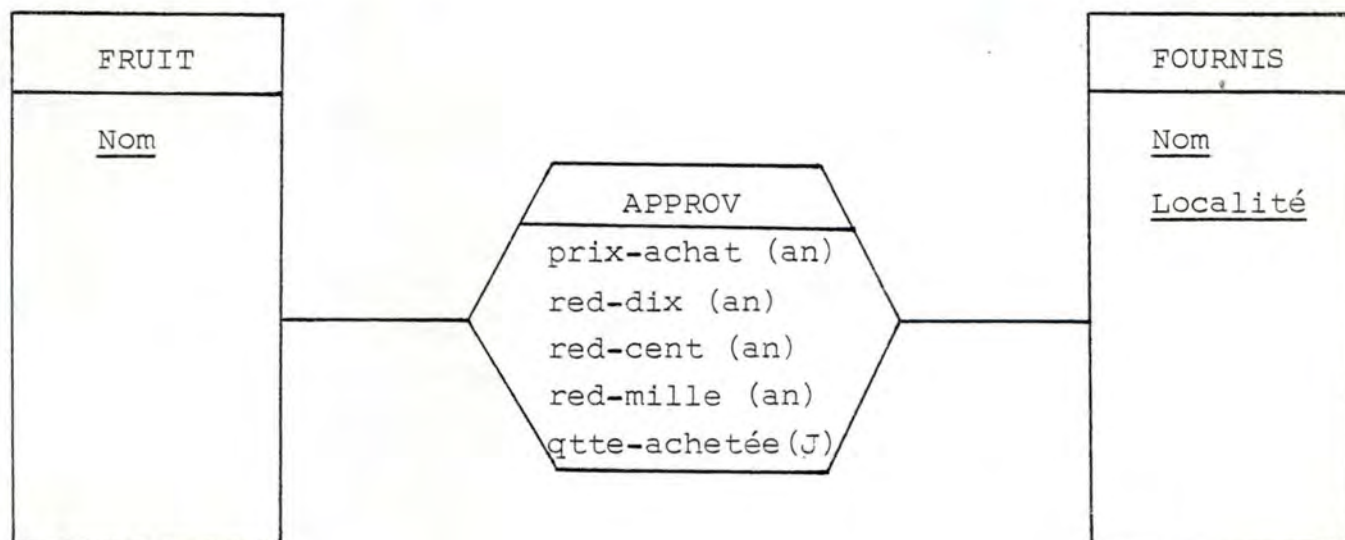
UTIL: P

SYST: Période d'évaluation ?

JOUR ? (J), MOIS ? (M), ANNEE ? (A)

UTIL: J

SYST: Ce schéma représente-t-il bien ce que vous voulez décrire ?



UTIL: OUI

SYST: Suite des spécifications de l'attribut "QTTE-ACHETEE"
du T.A. "APPROVIS." DEFINITION?

UTIL: Quantité achetée d'un certain fruit chez un
fournisseur à une certaine date.

SYST: SYNONYME ?

UTIL: < RETURN >

SYST: DECOMPOSABLE ? (O) (N)

UTIL: N

SYST: Obligatoire (OB) ? optionnel (OP) ?

UTIL: OB

SYST: SIMPLE (S), REPETITIF (R) ?

UTIL: S

SYST: FORMAT ?

<type caractère><; nombre caractères>

<; nombre caractères décimaux>

type caractère = A (Alphanumérique-Alphabétique)
N (Numérique)

UTIL: N;4

SYST: Unité de mesure ?

UTIL: SANS

SYST: DOMAINE DE VALEURS ?

UTIL: < RETURN >

SYST: spécifications complètes

II .3.3. SYSTEME D'EQUATIONS.

Le modèle correspondant à ce problème sera déclaré de la manière suivante:

<u>MODELE</u>	COUT-KILO-FRUIT-EVALUATION-ANNUELLE.
<u>DEFINITION</u>	Pour un fruit donné, le modèle estime le coût du kilo à partir des achats effectués chez les différents fournisseurs pendant une période d'un an. Le coût du kilo est estimé en tenant compte des réductions octroyées lors de chaque achat. Le coût estimé est valide pour une période d'un an.
<u>RESULTAT</u>	COUT-KILO(FRUIT/ANNEE)
<u>ARGUMENT</u>	FRUIT, ANNEE
<u>DEBUT</u>	
⋮	voir équations page suivante.
<u>FIN.</u>	

II .3.4. STRUCTURES DES TABLEAUX.

La figure 17 représente une structure de tableaux susceptible d'être généré par ce modèle.

Toutes les informations ne figurent pas sur ce tableau: les valeurs qui devraient être introduites en colonne F proviennent des équations du modèle précédent.

Chaque case de la colonne F (libellée "CALC MOD.1") correspond à un "sous-tableau".

DEBUT

$\text{COUT-KG}(\text{FRUIT/ANNEE}) = \text{MONTANT-PAYE}(\text{FRUIT/ANNEE}) / \text{QTTE-ACHETEE}(\text{FRUIT/ANNEE}) .$

$\text{QTTE-ACHETEE}(\text{FRUIT/ANNEE}) = \text{SOM}(\text{FOURN} [\text{APPROV}] \text{FRUIT}; \text{QTTE-ACHETEE}(\text{FRUIT}, \text{FOURNISSEUR/ANNEE}) .$

$\text{QTTE-ACHETEE}(\text{FRUIT}, \text{FOURNISSEUR/ANNEE}) = \text{SOM}(\text{JOUR} [\text{CALENDRIER}] \text{ANNEE}; \text{QTTE-ACHETEE}(: [\text{APPROV}]/\text{JOUR})) .$

$\text{MONTANT-PAYE}(\text{FRUIT}, \text{ANNEE}) = \text{SOM}(\text{FOURN} . [\text{APPROV}] \text{FRUIT}; \text{MONTANT-PAYE}(\text{FRUIT}, \text{FOURNISSEUR/ANNEE}) .$

$\text{MONTANT-PAYE}(\text{FRUIT}, \text{FOURNISSEUR/ANNEE}) = \text{SOM}(\text{JOUR} [\text{CALEND.}] \text{ANNEE}; \text{MONTANT-PAYE}(\text{FRUIT}, \text{FOURN} . / \text{JOUR}) .$

$\text{MONTANT-PAYE}(\text{FRUIT}, \text{FOURNISSEUR/JOUR}) = \text{RESULAT} \text{ PRIX-TOTAL-PAYE}(\text{FRUIT}, \text{FOURNISSEUR}) .$

DU MODELE PRIX-ACHAT-AVEC-REDUCTION.

AVEC PARAMETRE QTTE-ACHETEE = QTTE-ACHETEE(:[APPROV]/JOUR).

FIN.

A	B	C	D	E	F
<u>ANNEE</u>	<u>FRUIT</u>	<u>FOURNISS</u>	<u>JOUR</u>	<u>QTTE-ACHET.</u>	<u>MONTANT-PAYE</u>
ARGUM	ARGUM	INTR.AUT	INTR.AUT	INTR.AUT	CALC.MOD 1
		INTR.AUT	INTR.AUT	x	CALC.MOD 1
			INTR.AUT	x	x
			⋮	⋮	⋮
			INTR.AUT	INTR.AUT	CALC.MOD 1
i.			<u>TOTAL</u>	Σ	Σ
		⋮	⋮	⋮	⋮
		⋮	⋮	⋮	⋮
		INTR.AUT	INTR.AUT	x	x
		INTR.AUT	INTR.AUT	x	x
			⋮		
			INTR.AUT	x	x
j.			TOTAL	Σ	Σ
k.		TOTAL		$E_i + \dots + E_j$	$F_i + \dots + F_j$
COUT-KG	F_k / E_k				

Figure 17.

II .4 . EXEMPLE 3 .

II .4.1. EXPLICATIONS .

Le gestionnaire désire une estimation grossière du prix de revient de ses produits (coûts directs, sans tenir compte des frais généraux, d'usure de machine, etc...).

L'estimation se fera en tenant compte du coût des composants du produit (fruit, matières premières), du coût de l'emballage (contenant + étiquettes) et du coût de main d'oeuvre.

II .4.2. SYSTEME D'INFORMATION .

Le système d'information correspondant au problème est représenté dans la figure 18. Afin de faciliter la compréhension, nous décrirons brièvement chaque élément

II .4.2.1. LES TYPES D'ENTITE .

FOURNISSEUR, FRUIT: utilisés précédemment

PRODUIT:

Une occurrence de produit désigne une matière manufacturée dans l'entreprise utilisée dans un produit fini

ex.: CERISE SIROP

CONFITURE 4 FRUITS

identifié par "nom".

MAT.-IERE:

Désigne les composants (autres que les fruits) nécessaires à la fabrication des produits

ex.: SUCRE, ACIDE CITRIQUE,....

identifié par "nom".

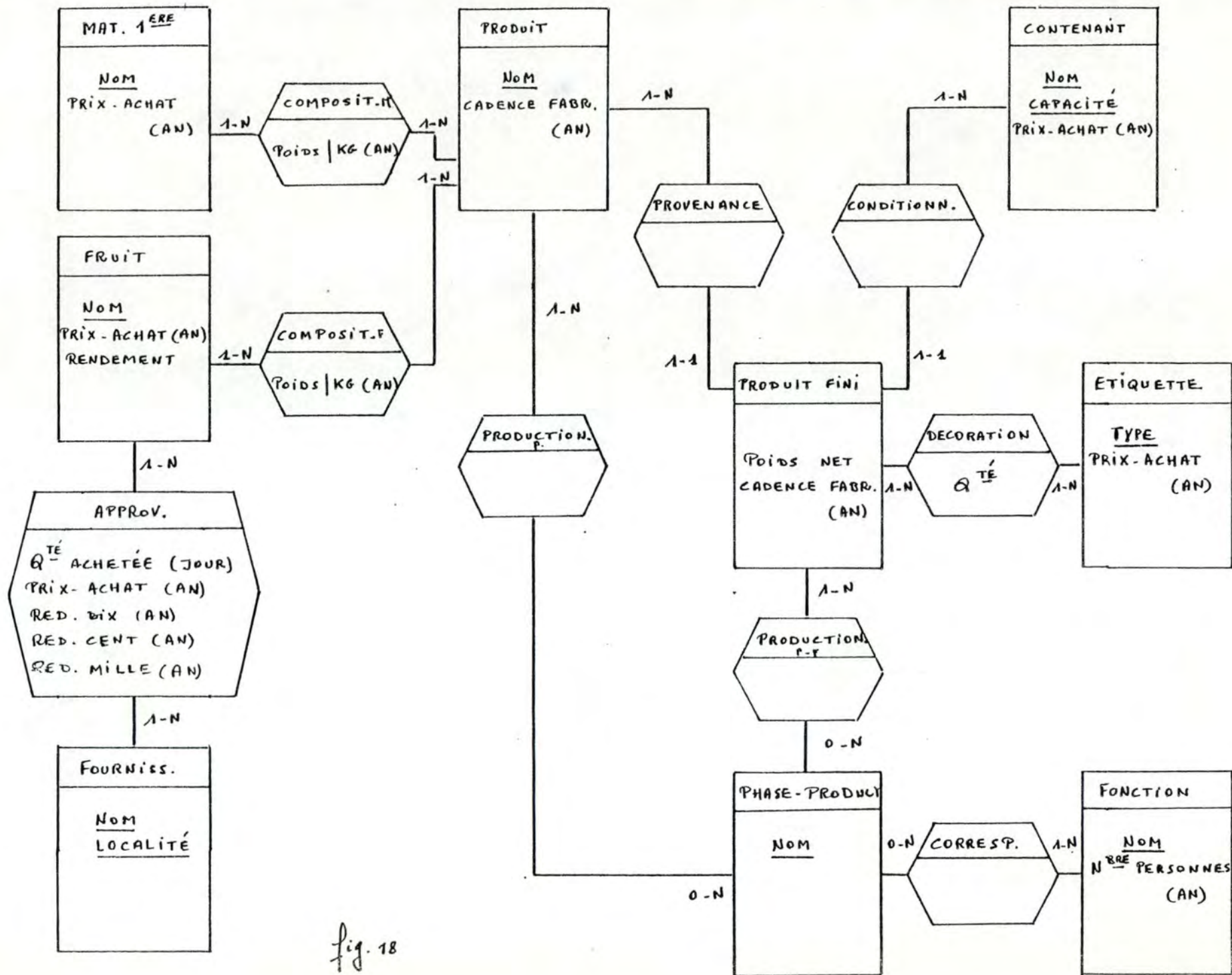


fig. 18

CONTENANT:

Un contenant est un récipient utilisé par l'entreprise pour conditionner ses produits.

ex.: WEEK, 1 L.; VERRE, 0,5 L.; conserve, 0,5 L.

PRODUIT-FINI:

Un produit-fini est un produit commercialisable identifié par le nom du produit qui lui est associé par le T.A. "provenance" et les noms et volume du contenant qui lui est associé via le T.A. "contenant".

ex.: CERISE SIROP, WEEK, 1 L.
 CERISE SIROP, CONSERVE, 1 L.
 CONFITURE ABRICOT, VERRE, 0,250 L.
 " " " ", " , 0,5 L.

ETIQUETTE:

Une étiquette est une décoration-papier appliquée sur le produit fini, identifiée par son type.

ex.: CLASSIQUE
 PETIT FORMAT

PHASE-PRODUCTION:

Désigne les différentes étapes d'un processus de fabrication

ex.: CUISSON
 ETIQUETTAGE

identifié par son nom.

FONCTION:

Une fonction est un poste de travail auquel sont assignés un certain nombre de personnes

ex.: VERIFICATION FERMETURE.

identifié par son nom.

II_4.2.2._LES_TYPES_D'ASSOCIATION.

COMPOSITION: (mat. lère, prod.), (fruit, prod.)

Une occurrence de cette association signifie qu'une matière première (respect. fruit) intervient dans la composition d'un produit.

PROVENANCE:

Une occurrence de cette association met en correspondance le produit fini et le produit à partir duquel il est fabriqué.

CONDITIONNEMENT:

Une occurrence de cette association met en correspondance le récipient conditionnant le produit fini et ce produit fini.

DECORATION:

Une occurrence de cette association signifie qu'une étiquette est utilisée pour décorer le produit fini.

PRODUCTION: (produit, phase_prod.), (prod.fini, phase_prod.)

Les occurrences de cette association mettent en correspondance les différentes phases de production par lesquelles passe le produit (respect. le produit.fini).

CORRESPONDANCE:

Les occurrences de cette association désignent les différentes fonctions correspondant aux phases de production.

II_4.2.3._LES_ATTRIBUTES.

Tout attribut identifiant est invariable avec le temps (les attributs identifiants sont soulignés dans la figure 18).

PRIX-ACHAT (MAT.PREM) :

Désigne le prix du kg de matière première. C'est un attribut variable avec le temps, de type instantané. Il est déclaré avoir une période de validité de type "année" (fixé en début d'année).

PRIX-ACHAT (FRUIT) :

Correspond au résultat COUT-KG(FRUIT/ANNEE) du modèle précédent. Il a donc une période de validité de type "année".

PRIX-ACHAT (. CONTENANT) :

Est le prix de l'unité de contenant attribut "instantané" déclaré avoir une période de validité de type "année".

PRIX-ACHAT(ETIQUETTE) :

Est le prix d'une unité "étiquette" attribut instantané
supposé avoir une période de validité de type "année".

RENDEMENT (FRUIT) :

Indique la proportion de fruit réellement utilisée dans la fabrication du produit (c-à-d après épluchage et dénoyautement des fruits); il est déclaré invariable dans le temps.

POIDS-KG [COMPOSITION] (MAT.IERE,PRODUIT)
(FRUIT,PRODUIT) :

Indique le poids de matière première (respect. fruit) intervenant dans la composition d'un kilo de produit. Cet attribut est déclaré invariable dans le temps.

CADENCE-FABRIQUATION (: PRODUIT) :

Désigne le nombre de kios de produit fabriqués en un jour, si la chaîne de fabrication est uniquement dédiée à ce produit.

N.B.: Bien que possédant une caractéristique dimensionnelle de type "jour", cet attribut est considéré comme attribut "instantané".

(A comparer, par exemple, à QTTE-FABRIQUEE(PRODUIT, JOUR) qui décrirait les différentes quantités de produit fabriquées par jour.)

Sa période de validité de valeur est également d'un an.

CADENCE-FABRIQUATION(:PRODUIT-FINI):

Désigne le nombre d'unités de produits finis fabriqués en un jour si la chaîne de fabrication est dédiée uniquement à ce produit.

(même remarque que précédemment).

POIDS-NET(PPRODUIT-FINI):

Est invariable avec le temps. En effet, le poids net dépend de la capacité du contenant; or, celle-ci fait partie de l'identifiant du produit fini.

QTTE-DECORATION :

Désigne le nombre d'étiquettes employées pour la décoration du produit fini. Cet attribut est déclaré invariable avec le temps.

NBRE-PERSONNES(FONCTION):

Désigne le nombre de personnes nécessaires à la réalisation de cette fonction.
Cet attribut est déclaré avoir une période de validité de type "année".

II .4.3. SYSTEME D'EQUATIONS.

Le modèle correspondant sera déclaré de la manière suivante:

<u>MODELE</u>	PRIX-REVIENT-UNITAIRE-PRODUIT FINI
<u>DEFINITION</u>	Le modèle estime grossièrement le prix de revient d'une unité de produit fini. Ce calcul ne tient compte que des coûts directs. Les valeurs des informations utilisées restent toutes constantes durant une période de type "année". Il en est donc de même pour le prix de revient calculé.
<u>RESULTAT</u>	PRIX-REVIENT (PRODUIT-FINI/ANNEE).
<u>ARGUMENTS</u>	PRODUIT-FINI, ANNEE.
<u>PARAMETRES</u>	SALAIRE-HORAIRE-PRODUCTION; NBRE-HEURE-TRAVAIL-JOUR.
<u>DEBUT</u>	
⋮	voir équations page suivante.
<u>FIN.</u>	

DEBUT

PRIX-REVIENT (PRODUIT-FINI/ANNEE) =
= COUT-EN-PRODUIT (PRODUIT-FINI) + COUT-EMBALLAGE (PRODUIT-FINI) + COUT-MAIN-D'OEUVRE (PRODUIT-FINI).
COUT-EN-PRODUIT (PRODUIT-FINI)
= (COUT-KG (PRODUIT) POUR PRODUIT [PROVENANCE] PRODUIT-FINI) x POIDS-NET (:PRODUIT-FINI).
COUT-KG (PRODUIT)
= COUT-MAT-PREM (PRODUIT) + COUT-FRUIT (PRODUIT).
COUT-MAT-PREM (PRODUIT) = SOM (MAT-PREM [COMPOSIT.-P] PRODUIT; COUT (MAT-PREM)).
COUT (MAT-PREM) = PRIX-ACHAT (:MAT-PREM/AN) x POIDS-KG (: [COMPOSITION-M/AN]).
COUT-FRUIT (PRODUIT) = SOM (FRUIT COMPOSIT.F PRODUIT; COUT (FRUIT)).
COUT (FRUIT) = PRIX-ACHAT (:FRUIT/AN) x POIDS-KG (: [COMPOSITION]/AN) / RENDEMENT (:FRUIT).
COUT-EMBALLAGE (PRODUIT-FINI)
= PRIX-ACHAT (:CONTENANT/AN) POUR CONTENANT [CONDITION] PRODUIT-FINI + COUT-ETIQUETTE (PRODUIT-FINI).
COUT-ETIQUETTE (PRODUIT-FINI) = SOM (ETIQUETTE [DECORATION] PRODUIT-FINI; COUT (ETIQUETTE)).
COUT (ETIQUETTE) = PRIX-ACHAT (:ETIQUETTE/AN) x QTTE (: [DECORATION]).
COUT-MAIN-D'OEUVRE (PRODUIT-FINI) = COUT-PRODUCTION (PRODUIT-FINI) + COUT-PRODUCTION-PRODUIT
(PRODUIT-FINI).
COUT-PRODUCTION (PRODUIT-FINI)
= NBRE-PERS-PRODUCT (PRODUIT-FINI) x SALAIRE-HORAIRE-PRODUCTION x NBRE-HEURE-TRAVAIL-JOUR
/ CADENCE-FABRIQUATION (:PRODUIT-FINI).

NBRE-PERS-PRODUCT (PRODUIT-FINI)
 = SOM (FONCTION [CORRESP.] PHASE-PRODUCTION [PRODUCTION-P-F] PRODUIT-FINI; NBRE-PERSONNE (:FONCTION/AN).
 COUT-PRODUCTION-PRODUIT (PRODUIT-FINI)
 = (COUT-PRODUCTION-KG (PRODUIT) POUR PRODUIT [PROVENANCE] PRODUIT-FINI) x POIDS-NET (:PRODUIT-FINI).
 COUT-PRODUCTION-KG (PRODUIT)
 = NBRE-PERS-PRODUCTION (PRODUIT) x SALAIRE-HORAIRE-PRODUCTION x NBRE-HEURE-TRAVAIL-JOUR
 / CADENCE-FABRIQUATION (:PRODUIT).
 NBRE-PERSONNE-PRODUCTION (PRODUIT)
 = SOM (FONCTION [CORRESP] PHASE-PRODUCTION [PRODUCTION-P] PRODUIT; NBRE-PERSONNE (:FONCTION/AN) .

FIN.

C O N C L U S I O N

Par ce travail, nous nous étions proposé d'évaluer la complexité d'exploitation d'un générateur de modèles d'aide à la décision.

Ce système a comme objectif de fournir à l'utilisateur certaines facilités :

- la possibilité d'utiliser un langage qui mettrait en évidence les concepts qu'il a l'habitude d'utiliser
- la suppression de manipulations fastidieuses.

Notre contribution a consisté en la conception de la Base de Connaissances du système et en un essai d'élaboration du langage de spécification de modèles.

Pour pouvoir conclure à l'aboutissement de ce système, il est nécessaire de se poser certaines questions :

- L'utilisateur pourra-t-il manipuler les concepts représentés dans le système d'information de la Base de Connaissances ?
Il semble à priori qu'il devra se faire seconder par un expert.
- L'utilisateur pourra-t-il, à partir d'un système d'information relativement complet et grâce à un support visuel, construire les modèles lui permettant de résoudre ses problèmes ?
- Le type de langage de formulation d'équations décrit n'est-il pas trop complexe ?

Au terme de ce travail, il semble qu'il est un peu tôt pour répondre à ces questions.

En effet aucun outil "interface utilisateur" n'a encore été spécifié. De plus, comme pour tous les S.I.A.D., seul le contact avec l'utilisateur et l'expérience permettront de conclure.

Une question d'un autre type peut également être posée :

- la mise en oeuvre d'un tel système n'est-elle pas trop complexe ?

La suite de ce travail pourrait donc consister en la spécification des outils "interface-utilisateur" ainsi qu'en la recherche d'une solution aux problèmes de mise en oeuvre qui semblent à priori complexes. Nous citerons entre autre :

- le mapping d'un schéma conceptuel vers un schéma de B.D.
- le mapping de la formulation d'équations formalisées en un programme de requête
- le mapping de l'expression d'équations formalisées en une description de la structure d'un tableau.

BIBLIOGRAPHIE.

- (1) M.S. GINZBERG, W.R. REITMAN, E.A. STOHR, eds : Décision Support Systems: Proceedings of the NYU Symposium on D.S.S. N.Y.-1981, North-Holland, 1982.
- (2) S.L. ALTER: Decision Support Systems: Current Practice and Continuing Challenges, Addison-Wesley, 1980.
- (3) H.G. SOL; ed. Processes and Tools for Decision Support, North-Holland, 1982.
- (4) J.L. BENNETT, ed. Building Decision Support Systems, Addison-Wesley, 1983.
- (5) G.FICK, R.H. SPRAGUE, eds : Decision Support Systems: issues and Challenges, Pergamon Press, 1980.
- (6) P.G.W. KEEN, M.S. SCOTT-MORTON: Decision Support Systems: an Organizational Perspective, Addison-Wesley, 1978.
- (7) R.H. BONCZEK, C.W. HOLSAPPLE, A.B. WHINSTON: Foundations of decision Support Systems, Academic Press, 1981.
- (8) M.S. GINZBERG, E.A. STOHR: DSS: Issues and Perspectives, pp 9-32 dans (1).
- (9) G.A. GORRY, M.S. SCOTT-MORTON : A Framework for Management Information Systems, Sloan Management Review, 13 (1) pp 55-70.
- (10) J.L. BENNETT : Analysis and Design of the User Interface for Decision Support Systems, pp 41-64 dans (4).

- (11) B. SHNEIDERMAN : Direct Manipulation: a Step beyond Programming Languages, pp. 57-69, IEEE Computer, August 1983.
- (12) C. SMITH et al. : Designing the STAR user Interface, pp. 272-304, Byte, vol. 7, N° 4, Apr. 1982.
- (13) M.M.ZLOFF: Office-by-Example: A Business Language that unifies DATA and Word Processing and Electronic Mail, pp. 272-304, IBM syst. j., vol.21, n°3.
- (14) M. SENKO : Foral LP for Diam II - Foral with Light Pen - A Language Primer, Research Report, IBM Research Center RC 6328, 11/12/76, 1976.
- (15) P.G.W. KEEN : Decision Support Systems: a Research Perspectives, pp. 23-44 dans (5)
- (16) F.P. BROOKS : The Mythical Man-Month, reading, Massachussets, Addison-Wesley.
- (17) T.H. JOHNSON, J.C. COURBON, B. OUDET : Conception et Mise en oeuvre des Systèmes d'Information: l'Approche Evolutive, Congrès AFCET "Modélisation et Maîtrise des Systèmes", Versailles, nov. 1977.
- (18) J.C. COURBON, J. GRAJEW, J. TOLOVI : L'approche évolutive dans la mise en place des systèmes interactifs d'aide à la décision, Papier de Recherche IAE - Grenoble, n°78-02, Institut d'Administration des Entreprises, Université de Grenoble II, jan. 1978.
- (19) J.C. COURBON, J. GRAJEW, J. TOLOVI : Conception et mise en oeuvre des systèmes interactifs d'aide à la décision par une approche évolutive.
- (20) J. GRAJEW, J. TOLOVI : conception et mise en oeuvre des systèmes interactifs d'aide à la décision : l'approche évolutive, thèse de doctorat de 3è cycle, IAE Grenoble, octobre 1978.

- (21) CARLSON, Eric D., J. BENNET, G. GIDDINGS, P. MANTEY : The Design and Evaluation of an Interactive Geo-Data Analysis and Display System, IFIP Congress 74, pp 1057-1067, 1974.
- (22) R. H. SPRAGUE : A Framework for Research on Decision Support Systems, pp 5-22 dans (5).
- (23) R. H. BONCZEK, C.W. HOLSAPPLE, A.B. WHINSTON : The Integration of Data Base Management and Problem Resolution, International Journal of Information System, vol. 4, N° 2, 1979.
- (24) F. MANOLA : Data Base Technology in Decision Support Systems, an Overview, pp 69-93 dans (5).
- (25) J.P. THIRY : Spécifications pour un système de génération de modèles, Unpublished paper, Institut d'Informatique, F.N.D.P.-Namur, June 1983.
- (26) F. BODART : Introduction to Decision Support Systems (Preliminary Draft), First International Summer School on Multiple Criteria Decision Making, Costa Ionica - Sicily, Août 83.
- (27) F. BODART, Y. PIGNEUR : Conception assistée des applications informatiques, première partie: Etude d'opportunité et analyse conceptuelle, Masson, Paris, 1983.
- (28) P.P. CHEN : The Entity-Relationship Model-Toward a Unified View of Data, ACM Transactions on Data Base Systems, 1, 1976.
- (29) F. BODART, Y. PIGNEUR : Le modèle entité-association, pp 14-50 dans (27).

A N N E X E S

LOCALISATION TEMPORELLE

DE L'INFORMATION

TABLE DES MATIERES

I. INTRODUCTION.....	A1
I.1. LES B.D. HISTORIQUES.....	A1
I.2. LA MODELISATION DE L'UNIVERS DU DISCOURS.....	A3
I.3. L'EVOLUTION DE L'UNIVERS DU DISCOURS.....	A3
I.3.1. Evolution de la constitution de l'univers du discours.....	A3
I.3.2. Evolution des caractéristiques des cons- tituants de l'univers du discours.....	A6
I.4. CONCLUSIONS.....	A8
I.5. LES OPERATIONS PERMISES SUR UNE B.D. HISTORIQUE.....	A10
II. LOCALISATION TEMPORELLE DES CONSTITUANTS DE L'UNI- VERS DU DISCOURS ET DES VALEURS DE LEURS CARACTE- RISTIQUES.....	A11
II.1. LES TYPES D'ENTITE.....	A11
II.2. LES TYPES D'ASSOCIATION.....	A12
II.3. LES ATTRIBUTS.....	A16
III. FORMALISATION D'EQUATIONS GENERALE EN FONCTION D'ATTRIBUTS DONT LES VALEURS VARIENT AVEC LE TEMPS.....	A25

I. INTRODUCTION.

I.1. LES BASES DE DONNEES HISTORIQUES.

La plupart des bases de Données représentent une vue statique du monde réel qu'elle modélise.

Ces changements qui interviennent dans le monde réel sont répercutés dans la B.D. par des modifications sur les données correspondantes: les anciennes données sont "oubliées".

Les données enregistrées dans ces B.D. sont donc conventionnellement relatives à l'instant courant.

Il n'y a donc pas lieu de pouvoir les localiser dans le temps.

Par contraste, une B.D. HISTORIQUE doit pouvoir représenter une vue dynamique du monde réel qu'elle modélise.

Les changements qui interviennent dans le monde réel seront représentés dans la B.D., par l'établissement d'une nouvelle "description"; aucune donnée n'est oubliée.

Les données enregistrées dans ces B.D. étant relatives à différents "instants", il y a donc lieu de pouvoir les localiser dans le temps.

Une B.D. historique peut donc être consultée comme un ensemble de B.D. "statiques" organisées de manière cohérente (voir fig.A1).

Nous appellerons "Univers du discours" (U.D.) la partie du monde réel que la B.D. modélise.

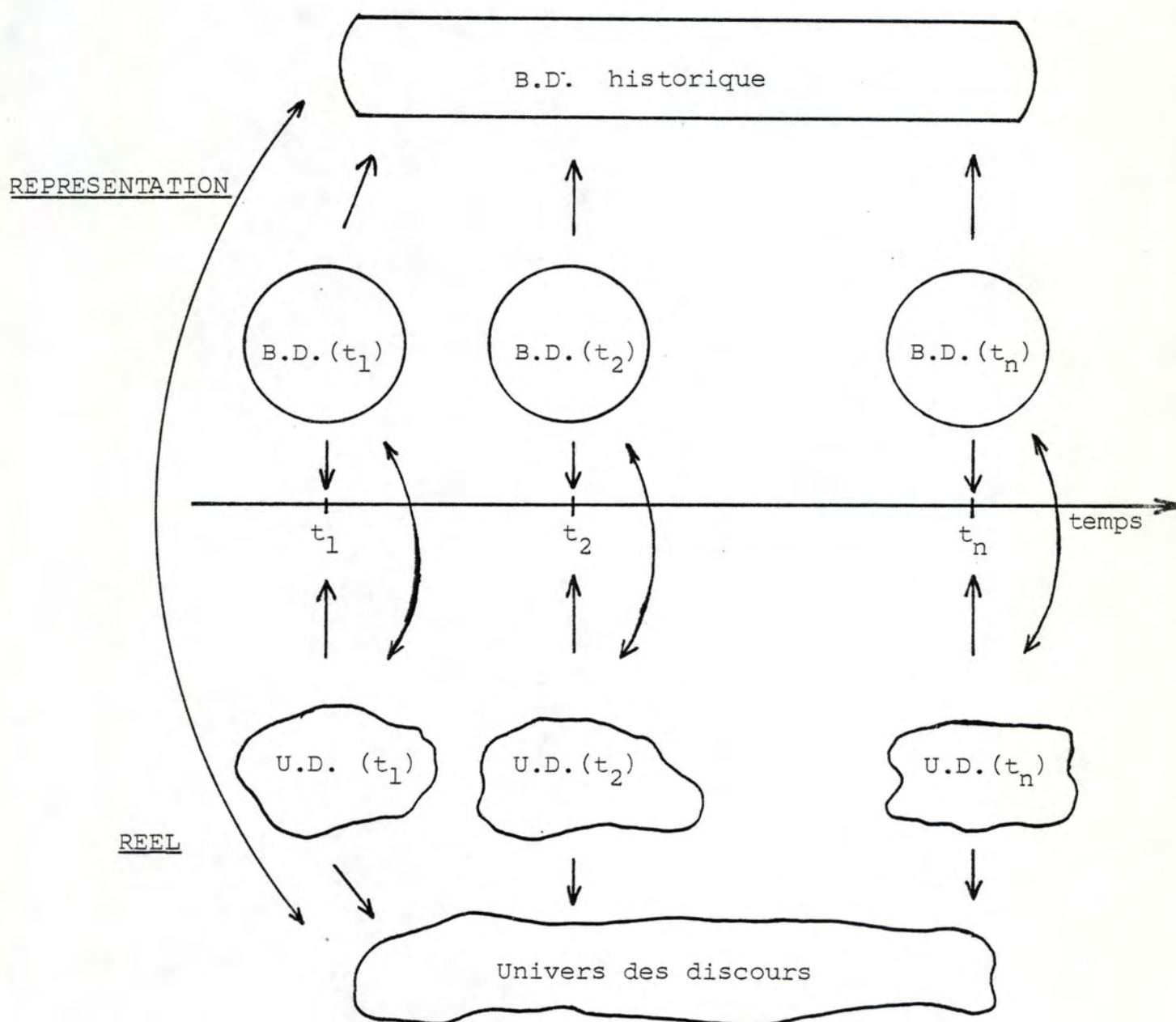


Figure A1.

I.2. LA MODELISATION DE L'UNIVERS DU DISCOURS.

Les "objets" (entités) de l'univers du discours sont regroupés en classes d'objets: types d'entité. La définition d'un T.E. décrit la condition d'appartenance d'un objet de ce type à l'univers de discours.

Les relations existants entre les objets de l'univers de discours sont regroupées en classes en relations définies sur des types d'objets: types d'association. La définition d'un T.A. décrit la condition d'appartenance à une association de ce type à l'univers du discours.

Les objets de l'U.D. et les relations entre ces objets possèdent des caractéristiques (attributs). Une entité est complètement décrite par la valeur de ses attributs. Une relation est complètement décrite par les objets qu'elle met en correspondance et la valeur de ses attributs.

I.3. L'EVOLUTION DE L'UNIVERS DU DISCOURS.

I.3.1. EVOLUTION DE LA CONSTITUTION DE L'UNIVERS DU DISCOURS.

A un instant donné, l'U.D. est constitué de certains objets et de certaines relations entre ces objets.

A un autre instant,

- de nouveaux objets peuvent faire partie de l'univers de discours (parce-qu'à cet instant, ils vérifient les conditions d'appartenance à l'U.D.), d'"anciens" objets ont disparu (parce-qu'ils ne vérifient plus la condition d'appartenance à l'U.D.).

- de nouvelles relations se sont créées, d'autres ont disparu.

ILLUSTRATION SIMPLE.

Supposons l'univers du discours défini, comme étant composé de 3 types d'objets (type d'entité): des objets de type EMPLOYE, des objets de type PRODUIT et les objets de type USINE.

Les relations pouvant exister entre les objets sont définies par deux types d'association:

- le type d'association TRAVAIL définie entre le T.E. USINE et EMPLOYE
- le type d'association FABRIQUE définie entre le T.E. USINE et PRODUIT.

Définitions:

T.E._employé: est considéré comme employé toute personne travaillant dans l'entreprise.

T.E._usine: une usine est un lieu de production des produits de l'entreprise.

T.E._produit: un produit est un article fabriqué par l'entreprise.

T.A._travail: une occurrence de cette association existe entre un employé et une usine si l'employé travaille dans cette usine.

T.A._fabrique: une occurrence de cette association existe entre une usine et un produit si le produit est fabriqué dans l'usine.

La figure A2 est une représentation simple de l'évolution de l'univers du discours.

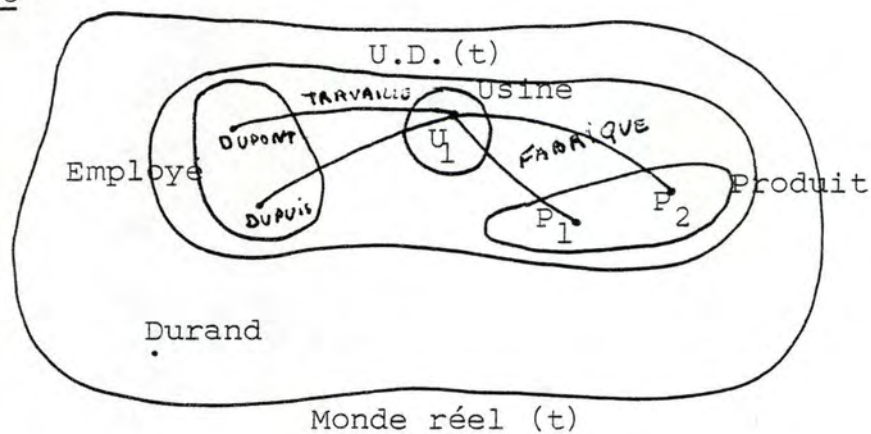
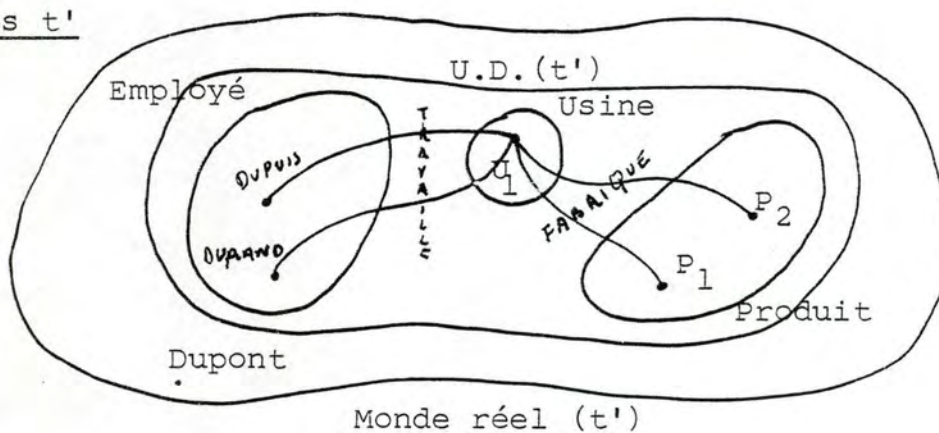
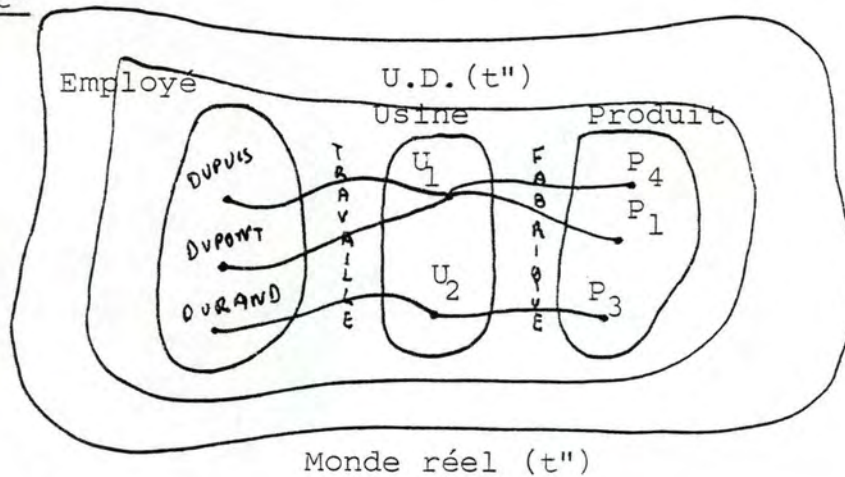
temps ttemps t'temps t''

Figure A2.

I.3.2. EVOLUTION DES CARACTERISTIQUES DES CONSTITUANTS DE L'UNIVERS DU DISCOURS.

Durant l'intervalle de temps pendant lequel un objet fait partie de l'univers de discours, ses caractéristiques peuvent changer.

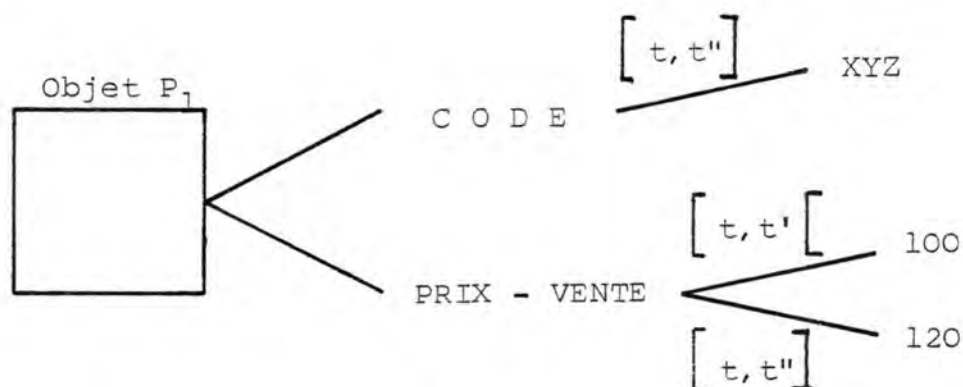
Illustration:

Soit l'objet P₁ représenté dans la figure précédente. Il fait partie de l'univers de discours, au moins pendant l'intervalle de temps $[t, t']$

Supposons cet objet caractérisé par un attribut CODE permettant de l'identifier et par un attribut PRIX-VENTE.

Durant l'intervalle de $[t, t']$ l'objet P₁ doit être complètement décrit par ses attributs.

Supposons une modification du prix de vente du produit P₁ au temps t' .



Un même objet P₁ peut donc être décrit par les valeurs (XYZ,100) et (XYZ,120) pendant sa "présence" dans l'univers du discours.

Durant l'intervalle de temps pendant lequel une relation entre objets fait partie de l'univers de discours, ses caractéristiques peuvent changer.

Illustration:

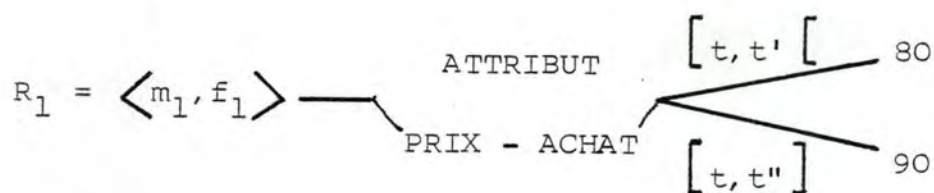
soient, les deux types d'entités: MATIERE-PREMIERE et FOURNISSEUR

Un T.A. ACHAT définit entre ces deux T.E., décrivant le fait qu'une matière première peut être achetée chez un fournisseur.

ce T.A. est caractérisé par un attribut PRIX-ACHAT dont une valeur indique le prix auquel le produit est acheté chez le fournisseur en question.

Supposons que la relation particulière $R_1 = \langle F_1, M_1 \rangle$ fasse partie de l'univers du discours pendant l'intervalle de temps $[t, t'']$, c-à-d. que la matière première m_1 peut-être achetée chez le fournisseur f_1 pendant l'intervalle de temps $[t, t'']$

Durant son appartenance à l'univers du discours, les valeurs de PRIX-ACHAT peuvent être modifiées.



I.4. CONCLUSIONS.

Une B.D. historique doit donc pouvoir représenter toutes les modifications survenues dans l'univers du discours:

- le début d'appartenance et la fin d'appartenance d'un objet à l'univers du discours.
- le début d'appartenance et la fin d'appartenance d'une relation entre objets de l'univers du discours.
- les modifications des caractéristiques des objets et des relations pendant leur appartenance à l'univers du discours.

Il est donc nécessaire de pouvoir localiser les objets, les relations et les valeurs des caractéristiques de ces objets et relations dans le temps.

Qu'est-ce que localiser dans le temps ?

Localiser dans le temps signifie faire référence à un point du temps ou à un intervalle d'une échelle de temps.

a) les échelles de temps.

Une échelle de temps est un ensemble ordonné de points du temps. Elle est définie à partir d'une durée privilégiée (une durée est un espace de temps), déterminée de telle manière que la considération d'une durée inférieure n'est pas pertinente. Cette durée privilégiée détermine les différents points du temps composant l'échelle de temps.

Un intervalle de temps défini sur une échelle de temps est un sous-ensemble des points du temps de cette échelle de temps et contient, outre ces deux points, tous les points du temps compris entre ces deux points. (cfr fig. A3)

b) la correspondance entre échelles de temps: le calendrier.

La correspondance entre échelles de temps est définie par un calendrier. Un calendrier définit les différents points du temps relatifs à une échelle de temps correspondant aux points du temps d'une autre échelle de temps (voir fig. A4).

Un point du temps d'une échelle de temps correspond donc à un intervalle d'une autre échelle de temps.

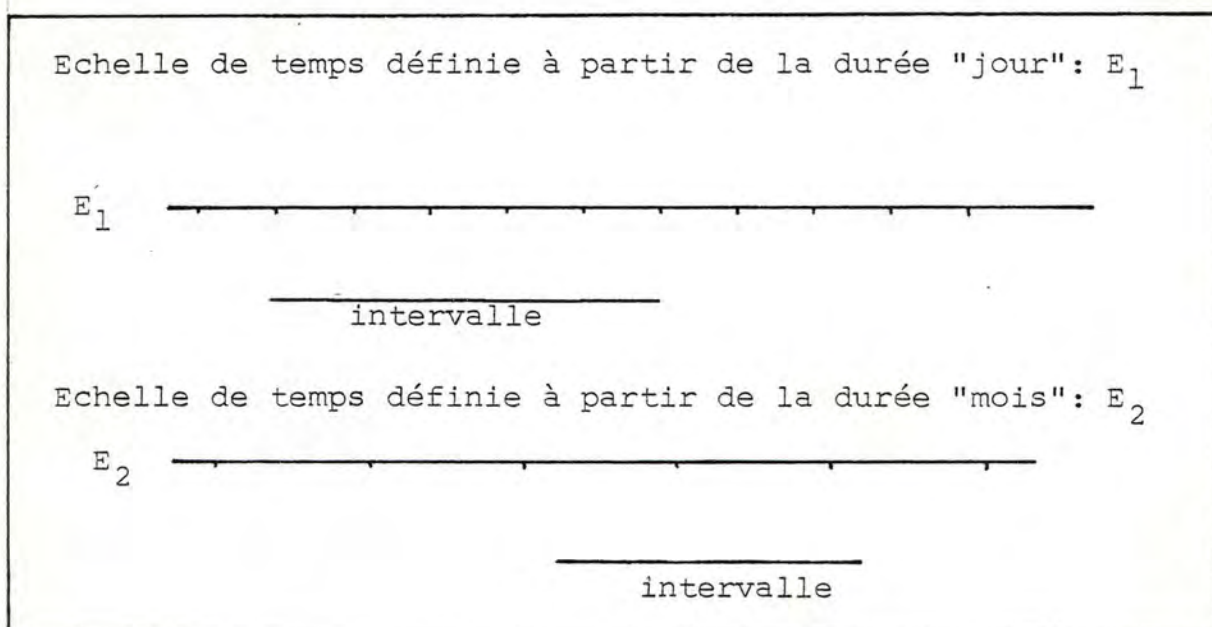


Figure A3

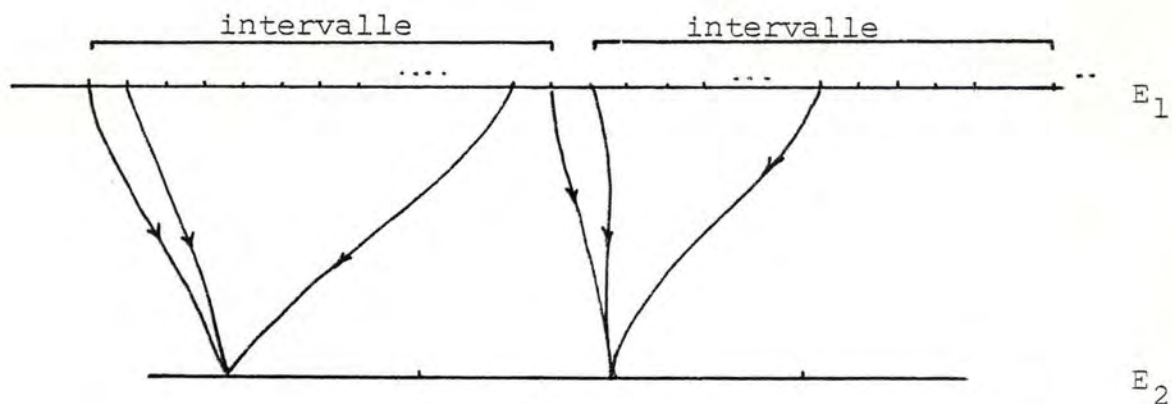


Figure A4

I.5. LES OPERATIONS PERMISES SUR UNE B.D HISTORIQUE.

L'histoire de l'univers du discours doit donc être enregistrée dans la B.D. Nous distinguerons deux types d'opérations: l'enregistrement et la correction.

Enregistrer signifie:

- soit, initialiser de nouvelles entités ou relations c'est-à-dire assigner une histoire initiale à leurs attributs.

- soit compléter les connaissances concernant les entités et relations en ajoutant de nouveaux "faits" à leur histoire déjà enregistrée.

Corriger signifie:

- modifier une histoire erronée en substituant une valeur correcte à une valeur incorrecte.

Il est important de distinguer l'histoire du monde réel (représentée dans la B.D. historique) de l'histoire des opérations effectuées sur ces représentations.

Ces deux histoires sont complètement indépendantes.

Le fait que le prix du produit P1 est "120" en janvier 83 peut avoir été enregistré le 01/03/83.

II. LA LOCALISATION TEMPORELLE DES CONSTITUANTS DE L'UNIVERS DU DISCOURS ET DE LEURS CARACTERISTIQUES.

La définition de toute "classe d'éléments" (T.E., T.A.) de l'univers du discours contient implicitement la condition de début d'appartenance et de fin d'appartenance des éléments de cette classe à l'univers du discours.

Ceci pose donc un problème de standardisation du procédé de localisation temporelle des éléments de l'univers du discours et de leurs caractéristiques.

Exemples:

Nous considérerons l'univers du discours comme étant une entreprise. Nous nous référerons à une échelle de temps définie à partir de la durée "jour".

II.1. LES TYPES D'ENTITE .

Soit le T.E. EMPLOYE,

défini par: est considéré comme employé toute
personne travaillant pour la firme.

caractérisé par: NO-EMPLOYE (identifiant)

NOM-EMPLOYE

ADRESSE-EMPLOYE

Soit le T.E. COMMANDE

défini par: une occurrence de cette entité correspond à un ordre de commande passé par un client à la firme.

caractérisé: NO-COMMANDE (identifiant)

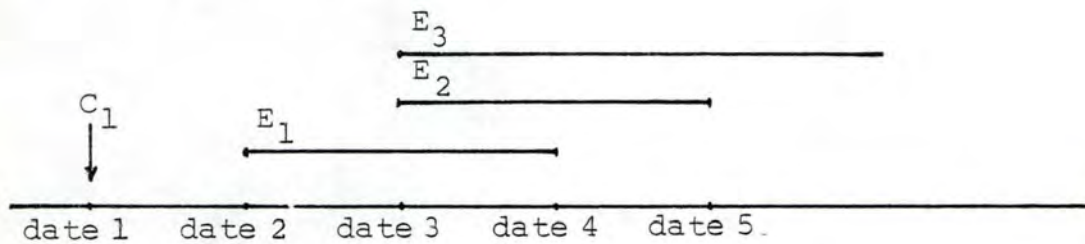
DATE-COMMANDE

MONTANT TOTAL COMMANDE

Par la définition et les caractéristiques associées au T.E. "COMMANDE", une occurrence de commande (ex.:C1) sera complètement localisée dans le temps (via date-commande). Les valeurs des caractéristiques qui lui sont associées ne varient pas (elles sont relatives à l'instant dat-commande").

Par contre, la définition du T.E. "EMPLOYE" ne fait que "suggérer" un moyen de localisation d'une entité de ce type dans le temps: ajouter deux attributs supplémentaires: DATE-ENGAGEMENT et DATE-FIN-ENGAGEMENT comme caractéristiques de ce T.E., afin de localiser une entité de ce type dans le temps.

Illustration:



EMPLOYE	N°	NOM	ADRESSE	D.E.	D.F.E.
	125	DUPONT	date 2	date 4
	128	DURAND	date 3	date 5
	131	DUPUIS	date 3	—

Ces deux attributs ont donc un statut privilégié puisque c'est par ce "mécanisme" que sera localisé le passage d'un employé dans l'entreprise.

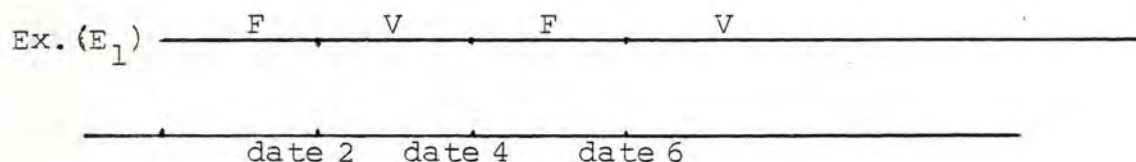
Remarque:

Comment tenir compte du fait que l'employé "DUPONT" qui a quitté la firme le "date 4" est le même employé que "DUPONT" qui a de nouveau été engagé le "date 6" ?

Création d'une nouvelle occurrence de EMPLOYE avec un même N° ?

Que devient la notion d'identifiant ? (de manière à sélectionner une seule occurrence, les deux dates définies précédemment doivent faire partie de l'identifiant).

Autre procédé de localisation temporelle: créer un attribut EXISTENCE pouvant prendre des valeurs booléennes. Pour pouvoir localiser une occurrence du T.E. EMPLOYE, l'histoire des valeurs de cet attribut serait enregistrée.



CONCLUSIONS.

Dans une B.D. "statique", lorsqu'une entité ne fait plus partie de l'univers de discours, les informations qui y sont relatives sont détruites (la B.D. est censée refléter l'instant courant).

Dans une B.D. historique, lorsqu'une entité ne fait plus partie de l'univers du discours, les informations qui y sont relatives sont conservées. Il est donc nécessaire

que pour chaque T.E. soit défini un "mécanisme" de localisation temporelle des éléments de ce type. En effet, il faut pouvoir déduire quelles sont les entités "actives" en un point (ou un intervalle) de temps.

ex.: quels sont les employés de 1981 ?

Quels sont les employés de 1983 ?

Il n'existe a priori, aucun "mécanisme" standard de localisation d'entités (le mécanisme doit être défini en fonction de la définition du T.E.).

II.2. LES TYPES D'ASSOCIATION.

Soit le T.A. "COMMANDE-CLIENT",

défini entre les T.E.: COMMANDE (défini comme précédemment) et CLIENT (personne ayant passé au moins un ordre de commande). (la définition de "CLIENT" suggère de localiser une entité de ce type dans le temps par un seul attribut "DATE-PREMIERE-COMMANDE").

défini par: une occurrence de cette association exprime le fait qu'un client a passé une commande.

Soit le T.A. "VENTE" définie entre les T.E. "PRODUIT" et "DEPARTEMENT"

défini par: une occurrence de cette association exprime le fait qu'un département vend un produit.

Une occurrence de l'association "COMMANDE-CLIENT" est complètement localisée dans le temps par l'occurrence du T.E. "COMMANDE" (via date-commande).

Par contre une occurrence de l'association "VENTE" sera localisé dans le temps, si nous définissons deux attributs: DATE-DEBUT-VENTE et DATE-FIN-VENTE, comme caractéristique de ce t.A.

Les occurrences produit: date1, date2 département k
 produit: date3, date4 département k
sont différentes.

Les dates permettant de localiser une occurrence d'association doivent être compatibles avec les dates permettant de localiser les entités mises en correspondance par cette association.

CONCLUSIONS.

Tout comme pour les entités, il n' existe pas de "mécanisme standard" de localisation d'associations. Le mécanisme doit être défini selon la définition des T.A. Il est cependant nécessaire de pouvoir localiser dans le temps les associations, de manière à déduire quelles sont les associations "actives" en un point (ou intervalle) du temps.

ex.: Quels ont les produits vendus par le département D1 en 1981 ?

Quels sont les produits vendus par le département D1
en 1983 ?

II.3. LES ATTRIBUTS.

La valeur d'un attribut d'une entité ou d'une association peut varier pendant la durée de vie de l'entité ou de l'association qu'il caractérise.

Nous pouvons a priori distinguer trois types d'attributs:

1) Les attributs "stables dans le temps".

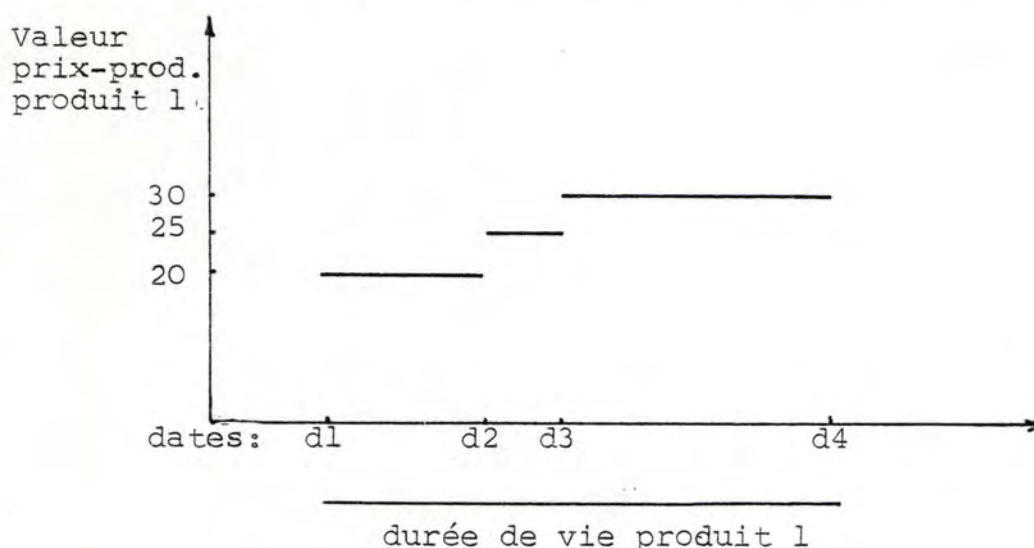
Ces attributs sont tels qu'une valeur qui leur est attribuée reste constante pendant toute la durée de vie de l'entité ou l'association qu'elle décrit (ex.: l'identifiant d'une entité doit être stable dans le temps).

2) les attributs "instantanés".

Une valeur attribuée à un tel attribut est susceptible de varier pendant la durée de vie de l'entité ou l'association qu'elle décrit. Une valeur peut lui être attribuée en tout point du temps (de la période de durée de vie de l'entité ou l'association).

exemple_1.:

Attribut "PRIX-PRODUIT" défini sur le T.E. "PRODUIT".



Pour localiser la valeur dans le temps, il est donc nécessaire de lui associer les dates pour lesquelles cette valeur est "active". (date-début activité, date-fin activité)
La période d'activité de la valeur correspond à sa période de constance.

(date-début-activité: DDA, date-fina-activité: DFA)

PRIX-PRODUIT	N°	PRIX	DDA	DDF
	P1	20	date 1	date 2
	P1	35	date 2	date 3
	P1	20	date 3	date 4
	P2	28	date 4	-

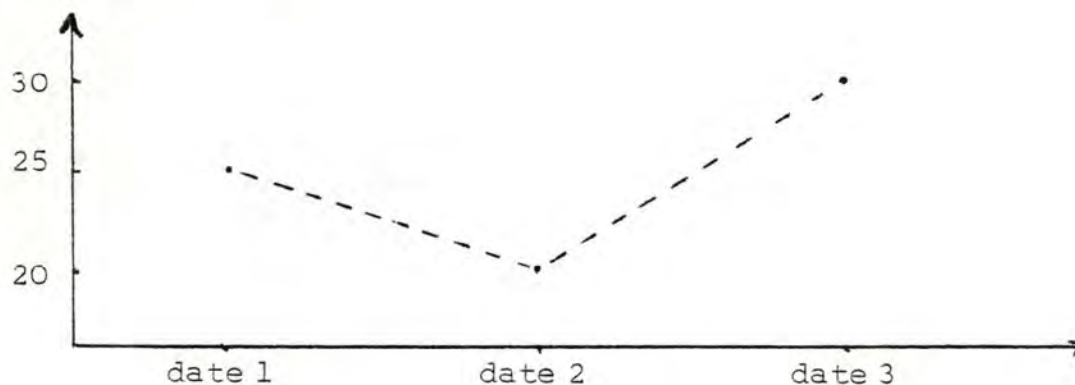
Une D.D.F. inconnue signifie que la valeur est toujours "active".

Exemple 2.:

ATTRIBUT "QTTE-EN-STOCK".

La valeur de cet attribut est susceptible de varier continûment. Il est donc nécessaire de déterminer des "instants caractéristiques" pour lesquels les valeurs seront enregistrées (nombre fini de valeurs).

L'histoire de cet attribut consistera donc en son ensemble de valeurs caractéristiques (correspondant aux instants caractéristiques) et en une fonction de dérivation, qui permettra de décrire, à partir des valeurs caractéristiques, les valeurs non données explicitement.



N°	QTEE STOCK	DATE
P1	25	date 1.
P1	20	date 2.
P1	30	date 3.

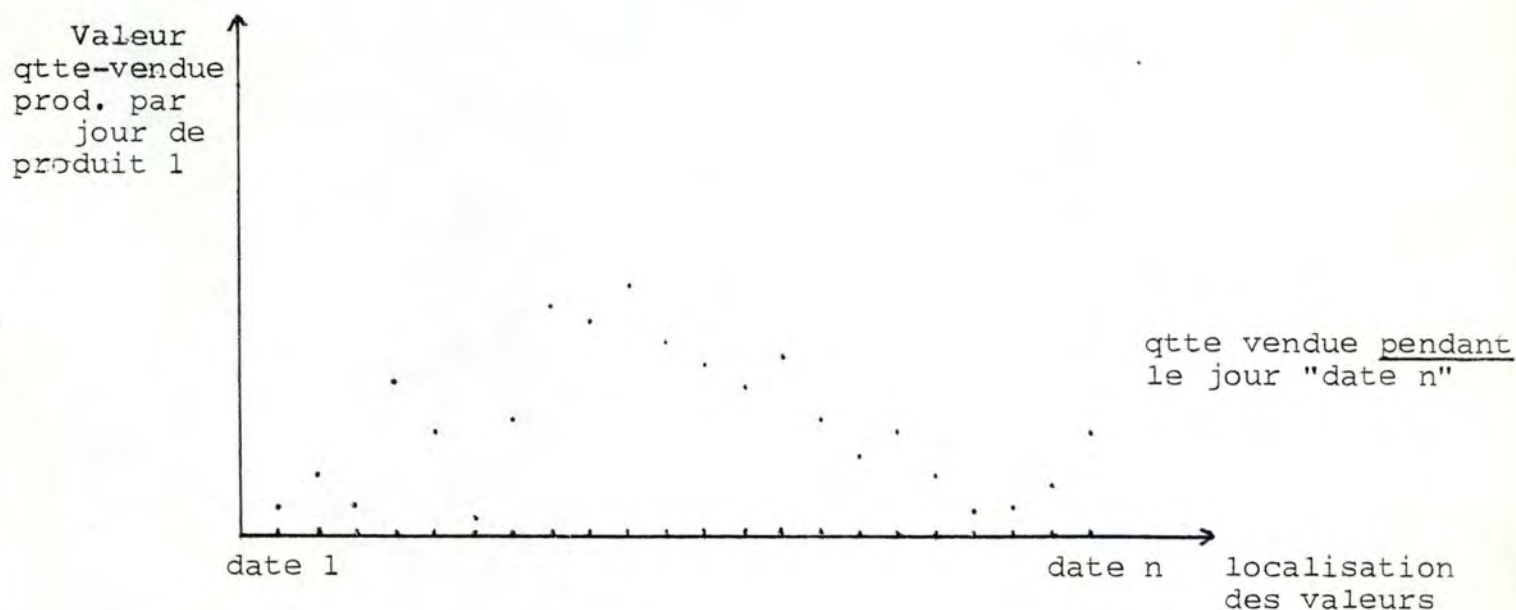
3) Les attributs "à période d'évaluation".

Ces attributs sont tels qu'une valeur ne peut être attribuée que relativement à une période d'évaluation

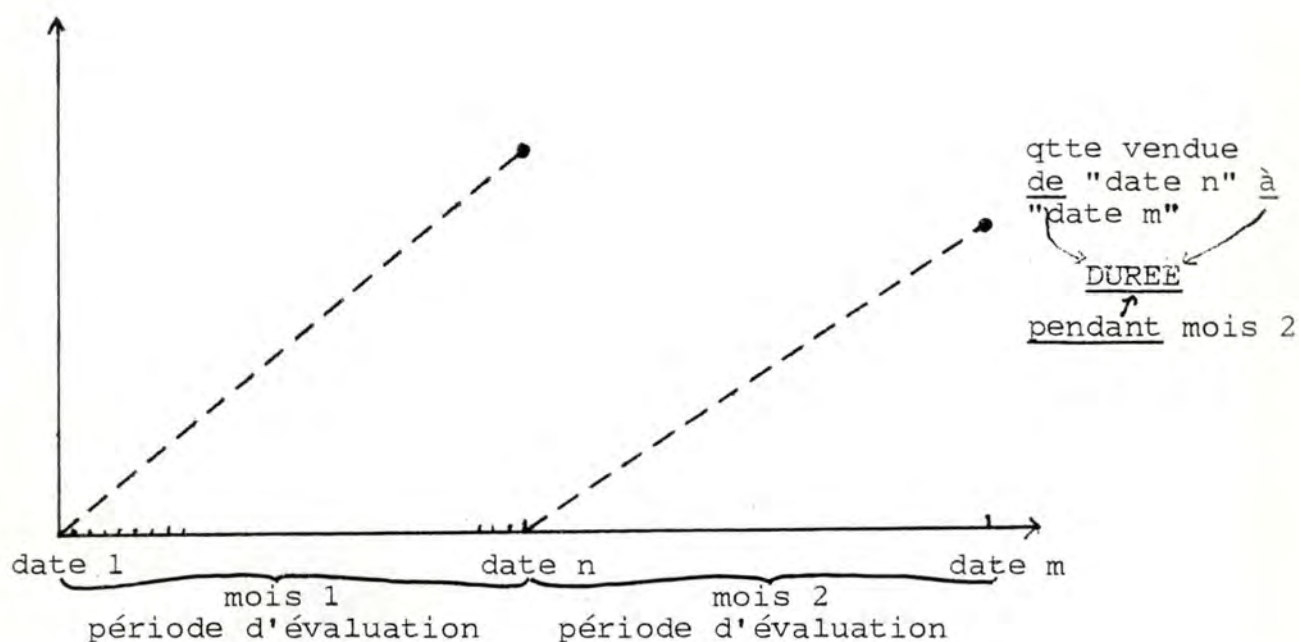
Les valeurs attribuées à ces attributs reflètent l'activité de l'entreprise pendant la période d'évaluation (notion de durée). Les valeurs associées à ces attributs doivent nécessairement varier dans le temps, puisqu'à chaque occurrence d'une nouvelle période du type de période d'évaluation (ex.: MOIS), une nouvelle valeur sera évaluée (remise à zéro en début de chaque occurrence du type période d'évaluation).

Exemple:

ATTRIBUT "QTTE-VENDUE-PRODUIT-PAR-JOUR".



ATTRIBUT "QTTE-VENTUE-PRODUIT-PAR-MOIS".



Dans ce cas, une valeur sera localisée par la période d'évaluation à laquelle elle est relative.

N°	QTTE-VENTUE-JOUR	DATE
P1	25	date 1.
P1	26	date 2.
P1	5	date 3.

Sur une échelle de temps de type "jour", une période de durée d'un jour ne peut être représentée que par une seule date.

N°	QTTE-VENTUE-MOIS	DATE-DEBUT-MOIS	DATE-FIN-MOIS
P1	700	01 01 81	31/01/81
P1	800	01/02/81	28/02/81
P1	320	01/03/81	31/03/81

Sur une échelle de temps de type "jour", une période d'évaluation de durée Δ jour doit être représentée par deux dates.

QUELLE ECHELLE DE TEMPS EMPLOYER POUR DECRIRE L'HISTOIRE DE VALEUR DES ATTRIBUTS ?

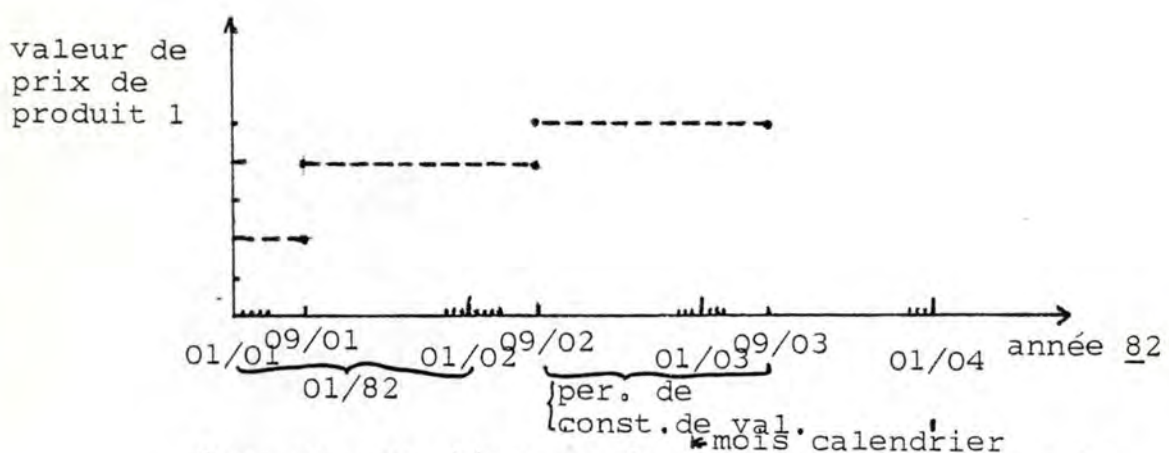
Localiser temporellement une valeur signifie faire référence à un point du temps d'une échelle de temps.

L'échelle de temps à employer dépend de la nature de l'attribut et du comportement de ses valeurs.

1) Les attributs "instantanés".

Les échelles de temps qui peuvent être employées pour localiser des valeurs de ce type d'attributs dépend de la localisation des modifications de leur valeur.

Exemple 1.



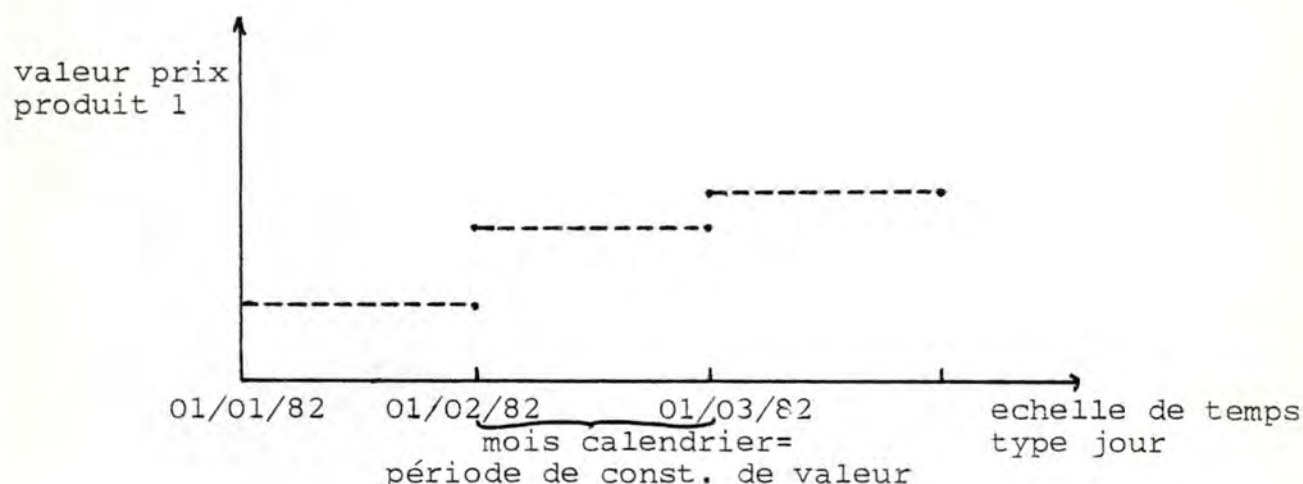
Dans ce cas, bien que la valeur reste constante pendant une durée d'un mois, (du 04.01.82 au 04.02.82;

du 04.02.82 au 04.03.82), l'échelle de temps à employer est celle définie à partir de la durée de "type jour".

En effet, on ne peut parler du prix du produit pour le mois 01/82 puisque ce point prend deux valeurs pendant ce mois.

Dans ce cas, une valeur sera localisée par une date.

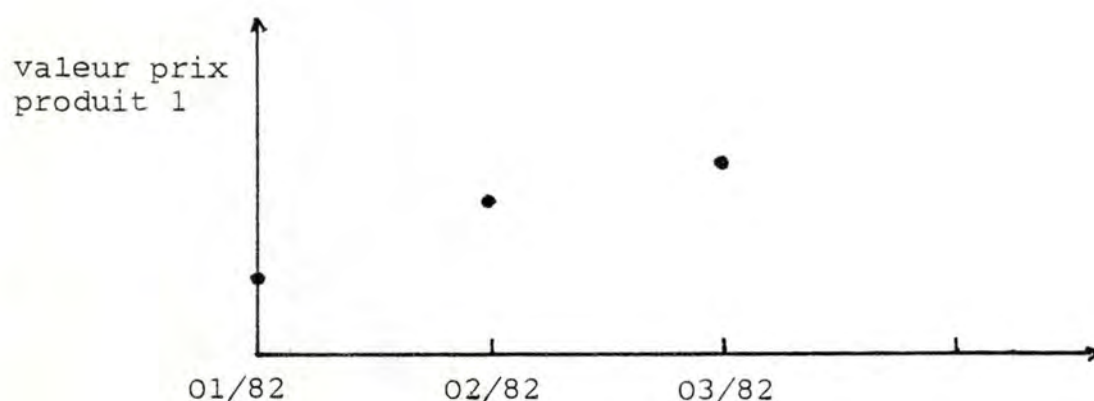
Exemple 2.



Dans ce cas, la valeur reste également constante pendant une durée un mois. Les modifications de valeurs sont localisées en des points particuliers. L'échelle de temps à employer peut-être celle définie à partir de la durée de "type mois".

On peut parler du prix du produit 1 pour le mois 01/82: ce prix ne prend qu'une seule valeur.

On peut donc passer d'une échelle de temps vers une autre.



L'échelle de temps à employer n'est donc pas fonction de la durée de constance d'une valeur, mais de la localisation de la période de constance de valeur.

Dans ce cas, une valeur peut être localisée par une occurrence de "mois".

2) Les attributs à période d'évaluation.

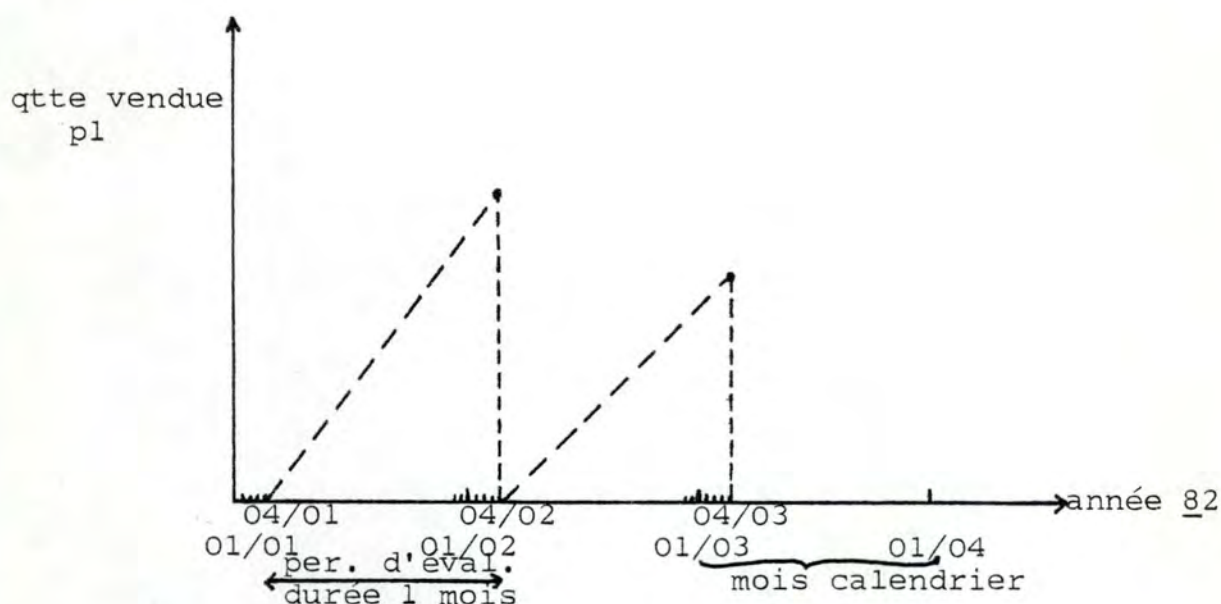
Une valeur attribuée à un tel attribut est relative à une période d'évaluation. Les échelles de temps pouvant être employées pour localiser une valeur de cet attribut dépend de la durée des périodes d'évaluation et de la localisation des périodes d'évaluation.

Il est évident que, lorsque la durée de la période d'évaluation est de type jour, l'échelle de temps à employer est celle définie à partir de la durée "jour" (puisque chaque jour (occurrence) une nouvelle valeur doit être évaluée).

Si la durée de la période d'évaluation est de type "mois", il faut que la période d'évaluation corresponde à

un "mois-calendrier" pour pouvoir employer l'échelle de temps définie à partir de la durée "mois".

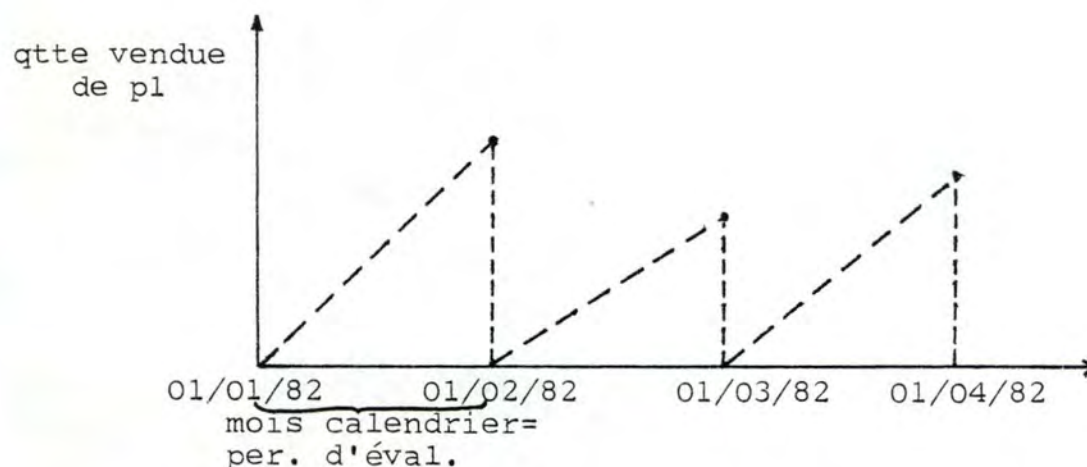
Exemple 1.



Dans ce cas, on ne peut parler de la quantité vendue du produit 1 pendant le mois X, mais de la quantité vendue du produit 1 pendant l'intervalle de temps 04-01-82; 04-02-82, par exemple. (valeur localisée sur une période "date-à-date").

Exemple 2.

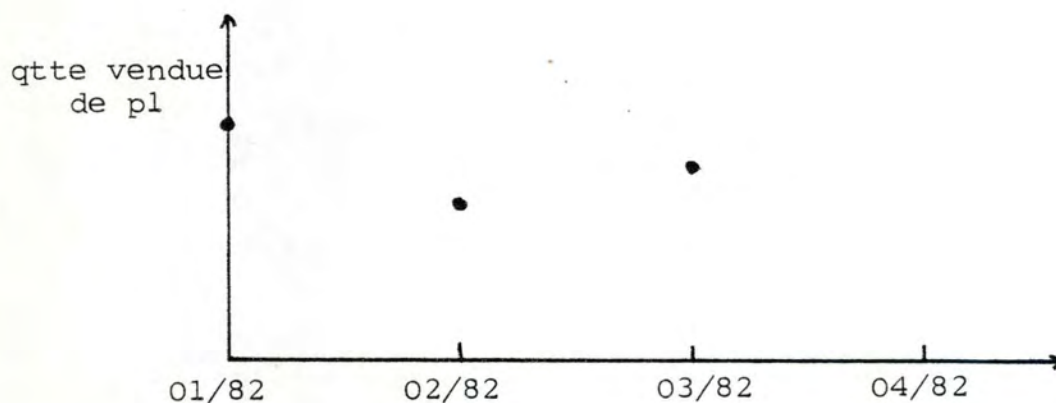
a)



Dans ce cas, on peut parler de la quantité vendue du produit 1 pendant le mois 01/82.

Le passage d'une échelle de temps vers une autre est permis.

b)



valeurs localisées par une occurrence de mois.

Selon le cas, une échelle de temps peut être plus appropriée qu'une autre puisqu'elle permet de faire référence à un seul point du temps de cette échelle (ex. 2b):

III. FORMULATION D'EQUATIONS GENERALES EN FONCTION D'ATTRIBUTS

DONT LES VALEURS VARIENT AVEC LE TEMPS.

Les équations générales devant permettre de calculer des informations à partir des informations contenues dans la B.D.H., devront être formulées en tenant compte des contraintes temporelles associées aux attributs:

- type de période d'évaluation des valeurs de l'attribut
- type de période pendant laquelle les valeurs restent constantes.

Considérons les équations générales:

$$\begin{aligned} & \text{BENEFICE (PRODUIT, PERIODE)} \\ &= \text{QTTE-VENTUE (PRODUIT, PERIODE)} \times \text{MARGE-BENEFICIAIRE (PRODUIT)} \\ & \text{MARGE BENEFICIAIRE (PRODUIT)} \\ &= \text{PRIX-VENTE (PRODUIT)} - \text{PRIX-REVIENT (PRODUIT)} \end{aligned}$$

Appliquées aux occurrences "P1" de PRODUIT
et "1984" de PERIODE,
ces équations générales deviennent:

$$\begin{aligned} & \text{BENEFICE (P1, 1984)} = \text{QTTE VENDUE (P1, 1984)} \times \text{MARGE BENEFIC. (P1)} \\ & \text{MARGE BENEFICIAIRE (P1)} = \text{PRIX-VENTE (P1)} - \text{PRIX-REVIENT (P1)} \end{aligned}$$

Elles pourront être exécutées si l'on connaît la
valeur du prix de vente du produit P1,
du prix de revient du produit P1,
de la quantité vendue du produit P1 pendant 1984.

Que se passe-t-il si pendant l'année 1984, le prix de vente du produit est modifié? (soit le 01-07-84)
 Pour pouvoir calculer le bénéfice réalisé sur le produit P1 pendant 1984, il sera nécessaire de décomposer cette équation en:

$$\begin{aligned}
 & \text{BENEFICE}(P1, 1984) \\
 &= \text{BENEFICE}(P1, [01-01-84, 30-06-84]) + \text{BENEFICE}(P1, [01-07-84, 31-12-84]) \\
 & \text{BENEFICE}(P1, [01-01-84, 30-06-84]) \\
 &= \text{QTE VENDUE}(P1, [01-01-84, 30-06-84]) + \text{MARGE BENEFICIAIRE}(P1) \\
 & \quad \text{"EN VIGUEUR" PENDANT []} \\
 & \text{MARGE BENEFICIAIRE}(P1) \text{ EN VIGUEUR PENDANT[]} \\
 &= \text{PRIX-VENTE}(P1) \text{ EN VIGUEUR - PRIX-REVIENT}(P1) \text{ EN VIGUEUR} \\
 & \text{BENEFICE}(P1, [01-07-84, 31-12-84]) = \dots \\
 & \quad \vdots
 \end{aligned}$$

Que se passe-t-il dans ce cas si la quantité vendue du produit P1 n'est connue que sur une période d'évaluation de type "année"?

De la valeur connue devrait être déduites les deux valeurs nécessaires au calcul ci-dessus.

etc...

Tous ces problèmes nous conduisent à imposer des restrictions:

- Nous ne considérerons que des types de période CALENDRIER (pas de période de date à date).
- Les équations générales seront exprimées de manière à respecter les contraintes de constance de valeur et de période d'évaluation de valeur.

Exemple:

Supposons qu'une quantité vendue de produit soit évaluée sur une période de type "mois" (quel que soit le produit),

que les prix de vente et prix de revient des produits restent constants pendant une période de type "mois" (quels que soient les produits).

Les équations générales correspondant au système cité précédemment seront formulées de la manière suivante:

$$\text{BENEFICE}(\text{PRODUIT/MOIS}) = \text{QTTE-VENTUE}(\text{PRODUIT/MOIS}) \times \text{MARGE-BENEFICIAIRE}(\text{PRODUIT/MOIS}).$$

$$\begin{aligned} \text{MARGE BENEFICIAIRE}(\text{PRODUIT/MOIS}) \\ = \text{PRIX-VENTE}(\text{PRODUIT/MOIS}) - \text{PRIX-REVIENT}(\text{PRODUIT/MOIS}) \end{aligned}$$

Si l'on veut pouvoir calculer un bénéfice réalisé sur une période de type "année", on ajoutera à ces équations l'équation

$$\begin{aligned} \text{BENEFICE}(\text{PRODUIT/ANNEE}) \\ = \text{SOM}(\text{MOIS} [\text{CALENDRIER}] \text{ANNEE}; \text{BENEFICE}(\text{PRODUIT/MOIS})) \end{aligned}$$

où la somme est faite sur tous les mois de l'année (relativement au calendrier).

Les équations générales ainsi formulées respectent donc les contraintes de constance de valeur et de période d'évaluation. Ces équations pourront être exécutées puisque toutes les informations nécessaires à leur exécutions sont connues dans la B.D. historique et que les différents types d'information font référence à la localisation temporelle employée (jour, mois).

De manière à pouvoir formuler les équations générales en respectant les contraintes temporelles associées aux attributs, il est indispensable de connaître le type de période sur lequel les valeurs sont évaluées et le type de période pendant lequel les valeurs restent constantes. Cette information doit être accessible à l'utilisateur.